Calhoun: The NPS Institutional Archive

Theses and Dissertations | Thesis Collection

1996-12

Cost/benefit analysis of interactive courseware

Himmelberg, Laura J.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/31988

http://www.nps.edu/library

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

COST/BENEFIT ANALYSIS
OF
INTERACTIVE COURSEWARE

by

Laura J. Himmelberg

December, 1996

Thesis Co-Advisor: Gordon Louvau

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 8

19970625 050

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December, 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE COST/BENEFIT ANALYSIS OF INTERACTIVE COURSEWARE

5. FUNDING NUMBERS

6. AUTHOR(S) Himmelberg, Laura J.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This purpose of this thesis is to present a cost/benefit analysis of different Interactive Courseware (ICW) packages available for use at Fleet Anti-Submarine Warfare Training Center (FLEASWTRACEN), San Diego, CA.

Data was gathered from interviews with the current users of ICW software at FLEASWTRACEN and at Goodfellow Air Force Base in Texas, interviews with companies that develop authoring systems, and information available on the Internet. Data was also gathered from information received from software companies and reports and memorandums available at FLEASWTRACEN. The goal of this thesis was to identify a cost-effective product that not only met the current training requirements at FLEASWTRACEN, but also one that would be able to expand and grow to meet their future requirements.

The nature of the pricing of the ICW software did not allow for a "strict" cost vs. benefit analysis. Since the costs vary depending on both the capabilities of the software package and the economic benefits received by the customer (which are "soft" and difficult to measure), the analysis focused on identifying a product that required minimal investment in additional hardware and/or software requirements. Based on evaluation of the research data, recommendations are presented for future acquisition of ICW software.

14. SUBJECT TERMS Cost/Benefit Analysis, Computer-Based Training, Interactive Courseware, Web-Based Training

15. NUMBER OF PAGES 146

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

DTIC QUALITY INSPECTED 3

Approved for public release; distribution is unlimited.

COST/BENEFIT ANALYSIS OF INTERACTIVE COURSEWARE

Laura J. Himmelberg
Lieutenant, United States Navy
B.B.A., University of San Diego, 1992

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
December 1996

Author: [redacted]
Laura J. Himmelberg

Approved by: [redacted]
Gordon Bouvau, Co-Advisor

[redacted]
David Matthews, Co- Advisor

[redacted]
Doug Moses, Co- Advisor

[redacted]
Reuben Harris, Chairman
Department of Systems Management

ABSTRACT

This purpose of this thesis is to present a cost/benefit analysis of different Interactive Courseware (ICW) packages available for use at Fleet Anti-Submarine Warfare Training Center (FLEASWTRACEN), San Diego, CA.

Data was gathered from interviews with the current users of ICW software at FLEASWTRACEN and at Goodfellow Air Force Base in Texas, interviews with companies that develop authoring systems, and information available on the Internet. Data was also gathered from information received from software companies and reports and memorandums available at FLEASWTRACEN. The goal of this thesis was to identify a cost-effective product that not only met the current training requirements at FLEASWTRACEN, but also one that would be able to expand and grow to meet their future requirements.

The nature of the pricing of the ICW software did not allow for a "strict" cost vs. benefit analysis. Since the costs vary depending on both the capabilities of the software package and the economic benefits received by the customer (which are "soft" and difficult to measure), the analysis focused on identifying a product that required minimal investment in additional hardware and/or software requirements. Based on evaluation of the research data, recommendations are presented for future acquisition of ICW software.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

In today's demanding business world, management personnel must be able to balance the training needs of the employees and the organization against their ever-dwindling resources of time and money. As employees are increasingly expected to "do more with less," how does the Training Manager ensure that employees get the quantity and quality of training they need to be effective? In addition to being effective, they also need to minimize the cost of the training and the loss of productivity created from the employee's time spent away from the job. One solution to this dilemma is Computer-Based Training (CBT).

The Navy also faces the same obstacles as the business world with their share of shrinking resources. The Navy currently employs computer software to develop their training materials for various courses taught at the many training commands. The technology surrounding the development and implementation of software is constantly changing and advancing. New companies are continuously entering the marketplace to develop specific courseware material. This entrance creates competition and promotes

constant technological progress.

Computer-Based Training has been around since the late 1960's in some fashion. At first, CBT was the result of individual programmers working with computer code to create unique lessons. Over the years, commercial vendors recognized the increasing demand and began productizing tools to help simplify the production of CBT. There are now dozens of such authoring tools on the market.

In general, authoring tools can be thought of as "tool kits" to help automate the design, presentation, and management of a CBT lesson, much the way word processors simplify spell checking. Using authoring tools, difficult programming tasks for handling graphics, text, and data bases are often simplified to typing a few command lines or clicking a few buttons on the screen.

B. OBJECTIVE

The focus of this thesis is to provide a cost/benefit analysis for sole-source acquisition of Computer-Based Training (CBT) and Interactive Multimedia Technology materials used at Fleet Anti-Submarine Warfare Training Center (FLEASWTRACEN), San Diego, CA. This analysis is performed to provide a basis for decisions in identifying Interactive Courseware (ICW) materials that maximize

FLEASWTRACEN's attainment of its objectives and to distinguish those from the ones that increase cost with little or no return on investment.

C. PROBLEM STATEMENT

Based on interviews with FLEASWTRACEN personnel and a review of their current authoring system, the following information was obtained.

Currently, FLEASWTRACEN is utilizing Mandarin for Windows as the principal authoring tool and Mandarin Solo for Windows for coding acoustic data presentations such as waterfall displays. Mandarin is produced by Marconi Simulation of Atlanta, Georgia. Marconi utilizes the twin screen system described above. However, the twin screen system is not required or used with all authoring packages. With today's rapid technology changes and enhancements, Marconi may not be improving their software quickly enough to meet the increasing training needs and requirements not only at FLEASWTRACEN, but also Navy or DOD-wide.

A product that is more versatile and compatible with many of the current computer systems throughout the Navy may be more useful. As technology improves, the courseware must be able to adapt and expand to meet the changing training requirements. This adaptation must also apply to future

changes in Naval operations that result in modifications to training requirements.

Presently, Mandarin supports full-motion video. However, since the video display resolution is so high (1280 x 1024), a video window appears very small. This is because video is meant to run at a resolution of 640 x 480. In addition, the current Central Processing Unit (CPU) speeds are insufficient to run video at this high resolution. The CPU speeds are hardware deficiencies which are not improved with different authoring packages. However, this could be upgraded with the addition of a language-based processor such as UNIX, which is not compatible with Mandarin. Touch screen is available with Mandarin; however, a hardware upgrade is again required. The current graphics software package utilized at FLEASWTRACEN does not support 3-D graphics. Mandarin does not support voice recognition or the ability to modify the exercise on-the-fly throughout the course. These needs may be met through other software packages, but they may not be effective enough to justify the additional costs of not only the supplementary enhancements, but also of changing software packages in "mid stream." In other words, a different software package will have to be either inexpensive to justify redeveloping

courses already produced, or be compatible with the current Marconi software.

D. RESEARCH QUESTIONS

The primary research question is the following: **What courseware material provides the most economical benefit for training at FLEASWTRACEN while enhancing the quality of instruction?**

Subsidiary questions to be addressed in assessing the costs and benefits associated with the courseware currently being utilized are:

1. **What companies are plausible and competitive suppliers of courseware?** In order to be plausible, they must have a foresight for the future and perform continuous research to "keep up with the times." In order to be competitive, the companies must not only provide a product that is competitive in capabilities and price, but also provide technical support that is reliable and responsive.

2. **What are the capabilities and requirements for the courseware?** The capabilities of the courseware must meet the training requirements. Any extra capabilities, above the minimum requirements, must enhance training as well as be cost-effective. The courseware requirements will be set by the training demands, as well as the authors of both the

courseware and the training, in order to reduce their learning curve and be cost-effective.

3. **Is a product with built-in capabilities the most efficient and effective method of procurement?** Is it more cost-effective to purchase a software package that has built-in graphics, audio and video capabilities, or is it better to purchase a software package that is compatible with other Commercial Off-The-Shelf (COTS) products? What are the limitations of the built-in products? Do they meet the training requirements? Will they be outdated soon, or are they expandable?

4. **Does the current provider meet all of the requirements?** As technology improves, training requirements also change. Is the current provider keeping up with the newly developed enhancements?

5. **Is the software user-friendly?** Both the author and the student must find the software easy to use, or easy to learn. The ideal software will provide a decreasing learning curve relative to the level of flexibility it provides both the student and the author.

6. **What are the industry standards compared with the Navy standards?** If the Navy standards vary from industry standards, it may hinder the ability to find a company that

meets the Navy standards. If the Navy standards vary from the industry standards, is it cost-effective to change the Navy standards?

7. **Does the student recognize and appreciate enhancements to training afforded by competing concepts?** Will the enhancements affect the student's learning? Do the enhancements provide a more realistic simulation so that the students will retain more information and transition to the fleet with greater ease?

E. METHODOLOGY

Material for this thesis was obtained through a review of literature accumulated from the library and the Internet, along with information received through interviews. My first step was to identify a set of criteria in order to evaluate competitive companies. I then developed a set of criteria to evaluate candidate interactive courseware systems.

Next I proceeded to collect data on the characteristics of the various systems. After choosing seven companies, evaluating their systems based on the identified criteria and determining their effectiveness, I selected one company among the seven I reviewed. Based on the information gathered and completeness to which this company's product

met the criteria and the needs of FLEASWTRACEN, I chose CAST as a "winner".

F. ORGANIZATION OF THESIS

This thesis is divided into six chapters. This first chapter has provided a description of the problem and identified research questions to be addressed in the thesis.

Chapter II discusses the background of CBT and ICW. It also outlines some of the general advantages and disadvantages of CBT and ICW, as well as some specific benefits for FLEASWTRACEN and the Navy.

Chapter III explains some of the differences in authoring capabilities and lists some implementation considerations to assess prior to purchasing CBT or ICW. Additionally, the attributes of a CBT authoring tool and the current requirements and capabilities at FLEASWTRACEN are delineated.

A brief comparison of some of the top competing companies is performed in Chapter IV. Seven companies were evaluated and the comparison notes are included in this chapter.

An evaluation and analysis of the companies based on evaluation criteria determined from Chapters III and IV is provided in Chapter V. Some productivity considerations to

examine as well as the product selection are also discussed.

Chapter VI provides the conclusions and recommendations, along with some topics for further research.

II. BACKGROUND OF CBT AND ICW

The purpose of this chapter is to define some of the terminology used throughout the paper. It also describes the benefits, disadvantages and presentation formats, along with some general notes on CBT and ICW. This chapter concentrates primarily on CBT and ICW in general. A specific evaluation of the use of ICW at FLEASWTRACEN is discussed later in the paper. The material contained in this chapter is a consolidation of information obtained from various magazine articles, Internet web pages, and field research.

A. DEFINITIONS

The definitions contained in this section were gathered based on the current system utilized at FLEASWTRACEN, developed by Marconi. However, the first four terms are generic for all authoring systems. The final two apply to systems that use team leaders, subject matter experts, and quality assurance inspectors as described.

1. **Computer-Based Training** links multimedia technology with educational content and information that results in interactive training. CBT directly aids the instructor, eliminates labor, makes the instructor more effective and

focuses on students that genuinely need an instructor's assistance. It allows students to learn at their own pace, which, for many, is quicker than an instructor-taught course, and increases the amount of information that they retain. CBT also provides for consistent presentation of information. When properly developed and implemented, this new technology can help an organization: shorten training time; reduce training delivery and maintenance costs; increase participant motivation, achievement, and content retention; increase consistency in training content and delivery; improve access to information; and automate the tracking and measurement of participant progress.

2. **Interactive Courseware (ICW)** is the set of programs, graphics, files, procedure files, etc., that together, enable the computer to present the lessons of each course. Computer-Based Training is a subset of ICW along with Performance Support System (PSS), which will be described later. The courseware includes a shell and a set of authoring tools, or similar utilities that creates an environment for authoring.

3. **Authoring System** is the software used to develop ICW. Numerous companies develop a variety of authoring systems with various degrees of versatility and difficulty.

Many of these systems do not require the author to have programing knowledge or experience. The systems are built around a language or a data base structure. "Systems built on a data base structure use fixed pedagogical elements like icons of Icon Author as explained later in the paper. Language based systems must use some form of text editor to create the scripts written in that language (such as MSDOS through Windows or UNIX." (Moritz, 1996)

4. **Authors** are personnel who develop each lesson, employing the courseware shell and authoring tools provided in the ICW software package. They may utilize a twin screen system (two computer monitors) and a word processing package to develop each lesson. They "program" the software to display pictures, graphics, and text where required for the lesson. Through coordination with the graphic artists, any special graphics, that may be necessary for the lesson, are developed and implemented. Likewise, they write the audio script for the audio specialist, who then record the audio portion of the lesson.

5. **Team Leaders** review each lesson during specific phases of it's development to ensure course standards and good design practices are followed. They write their comments for changes to the authors in the courseware shell.

This enables the author to make any changes or corrections to the lesson at logical development milestones as to prevent unforeseen major rewrites.

6. **Subject Matter Experts (SME)** also review the lesson before it is submitted for final production to ensure the material presented in the lesson is correct.

Once the lesson has been approved for final production by the **Quality Assurance Inspectors**, the audio, graphics and text portions are added. The final product is typically a thirty to forty minute lesson segment of a full multi-week course.

B. GENERAL BENEFITS OF CBT AND ICW

There are numerous benefits of CBT and ICW. However, not every environment will realize all of the advantages that follow. A complete assessment of the particular training requirements and resources is required.

Computer-Based Training can be *highly interactive;* however, "the increasing use of the term ICW, which stresses the Interactive along with the Courseware, is being done because the 'old' CBT just wasn't that interactive (Moritz, 1996)." It simulates a personal tutor and explains the subject matter that is programmed through audio and video graphics technology. ICW encourages interaction from the

student and provides hands-on experience. Practice sessions, quizzes, and tests evaluate responses and provide instant feedback.

The courses are *self-paced* and *user-controlled.* They can be accessed anytime and at any location with a personal computer. The simulations and exercises are geared toward real-world use and are consistently presented each time the lessons are viewed. The courses are time-efficient. Some courses can be put on-line or on CD-Rom so that personnel do not miss work to attend classes, or so that the material may be presented at many different sites.

CBT and ICW provide *high performance at a low price.* This is accomplished by keeping the structure, content, and capability of ICW to a minimum, thus reducing development costs. "Even the simplest and most basic CBT, when properly applied, can significantly enhance the performance of the students (Moritz, 1996)." The courses can be utilized repeatedly on or from multiple computers simultaneously, but at individual paces. CBT and ICW are especially useful for geographically-dispersed training, which is similar to "Distance Learning". It is more cost-effective than incurring travel expenses to attend classes. They also afford training "on demand."

My research unearthed a variety of interesting statistics concerning the value of CBT:

1. CBT produces at least 30% more learning in 40% less time and at 30% less cost than traditional classroom teaching (Perelman, 1990).

2. Students retain 70% of what they learn through CBT vs. 10% retention for lecture (Lager, 1996).

3. CBT can reduce training costs by 50% while achieving an equal or higher quality of learning in 40-60% less time (Hall, 1996).

CBT is of particular value to the adult student because of its *flexibility*. Whether it is an adult employee needing training to increase his work skills for a current job or desired promotion, or an adult student working toward a higher educational degree, CBT is available when the student is, which eliminates scheduling problems. Many programs also offer a "book mark" feature that allows a student to leave a training module and return to the same place later. Therefore, if a student does not have time to complete the whole session at once, he can work at it incrementally. In addition, students may work at their own paces; traditional classroom instruction is often too fast or too slow for particular students. CBT offers complete privacy --

learners are free to repeat training modules as often as needed to become proficient, to ask questions, and to express opinions (when training is "on-line") without the fear of embarrassment that many students associate with classroom learning.

For the younger student, CBT and ICW offer features (such as audio, video, and/or interactivity) that retain a child's attention for a longer period of time compared to conventional classroom instruction. Additionally, instructors can utilize an ICW system to promote teamwork by dividing the class into teams and playing "learning games."

CBT and ICW offer *increased learning value over traditional classroom instruction.* This is primarily due to their interactivity. Whereas most classroom learning experiences are fairly passive (for example, listening to a lecture in a large group), ICW involves the learner at every step and gives the student immediate feedback. Many programs use sound in addition to graphics to "grab" the student's attention. Also, the student must interact with the instructor (the computer) regularly by answering and/or asking questions as the training progresses. By using more senses (sight, sound, touch), the student gains more from the training, and retention is enhanced. In addition, CBT

and ICW provide more consistent training than a human instructor could.

Although the statistics of CBT mentioned earlier do not agree on exact numerical values for each benefit, CBT is of particular value to the adult student. It enhances learning, cuts costs (in most instances), and reduces classroom time, besides producing other miscellaneous administrative benefits. "When you work with cost per student, you get different numbers than when you use cost alone. One course for use by only a few students might not be terribly beneficial when looked at on a cost/student basis, but still might be enormously beneficial when compared to the cost of losing a capital ship or airplane (or even a mid-sea collision). How much training costs are reduced depends on the type of training, the resources previously required to conduct the training, the number of students in a given period that require the training, and a host of other factors." (Moritz, 1996)

Depending on the exact situation, CBT can provide more cost-effective training than other traditional methods. Although the initial start-up costs may be a bit scary, CBT *can save money in many ways*:

1. Training is available where the employees are

located, thus no travel or hotel expenses are incurred.

2. CBT takes less time, thus minimizing an employee's lost productivity due to time away from the job.

3. Each CBT session can be used repeatedly for different employees as needed, whereas an instructor would have to be paid for each separate training session.

4. CBT can be designed to incorporate several different ability levels into one program.

5. CBT can be used to train employees who are geographically dispersed.

CBT provides other *miscellaneous benefits* as well. Most programs can track/report student information such as how much time a learner has spent in a program, test results, etc. This simplifies the administrative burden for the Human Resources/Training Department. Instead of having to maintain cumbersome "hard copy" records of training, the HR professional or instructor can simply maintain automated records that the computer compiles. In addition, using CBT can free up a supervisor's time as well. For example, if a certain supervisor is always called upon to provide training to employees on a particular topic, using CBT to train on that topic frees that employee to concentrate his full efforts on his job without having to allocate time to

prepare for and conduct training. Other "soft" benefits for CBT are: increasing a student's familiarity with the computer and/or telecommunications, encouraging time management skills (student/employee is responsible for allotting time for and scheduling "class attendance"), and improving literacy and written communication skills.

C. SPECIFIC BENEFITS OF CBT AND ICW EXPERIENCED BY THE NAVY AND FLEASWTRACEN

The benefits discussed in this section pertain to specifically the Navy and FLEASWTRACEN. The information was obtained through interviews with FLEASWTRACEN personnel, summaries and memorandums retained at FLEASWTRACEN. These sources contained historical savings already realized through the implementation of ICW and projections of future savings to the Navy.

Through the implementation of CBT, the total FY-95 annual savings for Chief of Naval Education and Training (CNET) Operating and Maintenance, Navy (OMN) fund was $2.08 million. These savings were realized through a $1.9 million savings in reduced student time on base and $.18 million savings in trainer operating costs. The savings achieved in the trainers are due to the reduced time the students spend on them because of the "button orientation" capabilities through CBT. The students have been familiarized with the

location and function of the different buttons through the on-screen displays and questions provided through CBT. Additionally, not having to procure or upgrade trainers saved $35 million through non-recurring costs to Program Sponsors. Additional savings due to the capability to export training to ships and Afloat Training Groups (ATG's) have yet to be determined. There is a projected annual savings of $4.88 million for FY-96 through FY-00. (Summary of Interactive Courseware Interim Status Report)

FLEASWTRACEN has currently realized savings of more than $1.3 million through reduced course length. A projected savings of more than $3.9 million is expected in future years due to course length reductions based on already-approved ICW technology applications (Moranville, 1996). The reduced course lengths send the students out to the fleet sooner so that the Navy can reap the benefits of each individual's training quicker. Also, the overhead per student is reduced since classroom time is reduced.

D. DISADVANTAGES OF CBT AND ICW

Although CBT and ICW offers many benefits over traditional training methods, it is not always the best method in every situation. Trainers need to be thorough in their research/planning to ensure the right training

technique is selected. Oftentimes, ICW may appear attractive because it is "high-tech" which is currently exciting and may be considered "cutting edge" or the "wave of the future." But, if a "low tech" solution is better-suited to the situation and/or more cost-effective, the trainer needs to be careful not to get carried away by the excitement of implementing CBT or ICW.

CBT and ICW may not be cost effective for very small or specialized/technical groups of employees. The cost of developing a computerized training program for a small group of technical employees may be higher than the return on investment. CBT and ICW is better suited to training situations where there is a large volume of participants and it is anticipated that the same training module could be used repeatedly.

Some forms of CBT and ICW are not easy to update. For example, training modules on a CD-ROM may take months to develop. If the information on the CD-ROM will be out-of-date by the time the module is completed, then this is obviously a poor choice of training method. "Traditional CBT" to be used on a stand-alone computer system should not be used to present training on subjects where the information is constantly changing. However, on-line

training programs are fairly easy to update and may be of value in this type of situation (on-line training will be discussed in more detail later).

CBT and ICW require that the employee/student be committed to receiving the training and that the *organizational culture encourage completion of the course.* Since CBT and ICW allow for individual flexibility, it would be easy for an employee in a demanding/hectic work environment to keep "putting it off until tomorrow." The solution to this problem, however, is organizational emphasis. For required training, the employee could be given a "must complete by" date. The culture of the company will play a major role in how employees perceive the training and whether or not they complete it.

In the past, *most CBT and ICW requires both literacy and language fluency.* Therefore, it would not be an effective training medium for any employees who are either illiterate or do not speak the language fluently enough to understand the training. However, more software companies are producing voice tracks in multiple languages so that the user can select the appropriate language. "CBT's intent is to teach language skills. Think about Sesame Street children's learning materials which are presented via CBT."

(Moritz, 1996)

CBT and ICW programs require that the *persons to be trained have easy access to a computer* and have *enough basic knowledge of a computer* to be able to participate in the training. This is ideal in an office setting, where nearly every employee has a desk-top computer. However, in a "field" environment, a learning center would have to be established that would provide the employees easy access (location and hours of operation) to the computers. Although "computer literacy" levels may need to be considered to run most CBT and ICW programs, employee skills must be analyzed before implementing an ICW program. Additionally, with the advancement of the computer age and addition of a mouse, minimal computer skills are now required. "Good ICW recognizes the incoming skill and education levels of the prospective student" (Moritz, 1996).

CBT programs require the host computer to have certain minimum capabilities. Before implementing a CBT program, it is essential to determine if the computers to be used for the training have the appropriate hardware and supplementary equipment to run the program. "The course objectives and the courseware that is developed to meet those objectives set the hardware requirements. CBT programs can be written

for various hardware suites." (Moritz, 1996) Although some programs do not require as "fancy" a computer as others, old or low-grade computer equipment will limit the options available. In addition, some benefits of CBT (sound, graphics, etc.) may be reduced if the computer system is not up-to-date enough to incorporate those capabilities. If your training needs require the use of audio, visual orientation, or accurate simulation, then the hardware must be more sophisticated to support these additional demands, and the CBT programs must be written to incorporate them.

The final disadvantage of CBT is that as a new technology, *development costs may be tough to justify to upper-level management.* In fact, a 1992 article in Corporate Computing stated that Training Managers from companies that had implemented multimedia CBT programs reported that their biggest hurdle was convincing top management of the cost-effectiveness of the initial investment in the system (Bandrowski, 1992). Additionally, the built-in bias of the older manager types who are "computer phobic" is difficult to overcome. These managers have not used a computer, do not want to learn, or do not believe they could learn. The benefits must be demonstrated and proven to these managers.

E. CBT PRESENTATION FORMATS

There are five major presentation formats for CBT: tutorials, drills, simulations, games, and tests. Each may be used individually or in combination with the other types depending on the objectives of the specific training program.

1. **Tutorials:** Tutorials are probably the most common format of CBT in use today. They follow a standard format of a) presenting information to the student, b) questioning the student about the information, c) judging/evaluating the student's response, and d) giving feedback to the student. Typically, the program will not allow the student to move on to new information until he demonstrates an understanding of the material covered thus far. Tutorials are most appropriate for presenting factual information, teaching rules and principles, and teaching problem-solving strategies.

2. **Drills:** Computer drills are mainly a method for practicing. They are ideal for any situation where fluency with the material is necessary such as in basic skills, foreign languages, spelling, and vocabulary. Drills can also be used to allow a student to practice applying rules such as the laws of search and seizure. The computer would

give the student a list of facts, and the student would apply the rules to that specific situation.

3. **Simulations:** Simulations provide a real-world type scenario where the student can actually apply the knowledge he has learned. The main advantage to simulations is that they are often safer and more cost-effective than actually performing the task, especially if the task concerns an expensive or dangerous piece of machinery or process. Simulations also increase the motivation of the students because they are more "real-life" than just answering questions in a tutorial or practicing a drill.

4. **Games:** Games are similar to simulations, but are generally more fun to do and may or may not mimic reality. There are several types of games: adventure games, arcade-type games, board games, gambling games, logic games, role-playing games, quiz games, and war games.

5. **Tests:** Computerized tests are designed to do two things: a) quiz the students on their knowledge of the material, and b) provide feedback to the student on his progress. The feedback is an important part of the learning process in that it can help fill the gaps in the student's knowledge and/or correct any misconceptions the student may have. Tests can also add a sense of competition that

typically promotes learning and retention. In addition, tests can be used before training to determine the student's knowledge level and then allow the computer to design a tutorial that will cover only the skills/information that the student is lacking.

F. NOTES ON CBT AND ICW

This section discusses delivery platforms, forms of ICW, and the most common tool used to develop different forms of ICW. This information is a compilation of items discussed in various magazine articles and interviews with users and developers of ICW. The delivery platforms addressed in this section are: 1) Stand-alone CBT Systems and 2) Web-Based Training/On-Line Instruction. Three forms of ICW are described: 3) Intelligent Tutoring Systems, 4) "Just In Time" Learning and 5) Performance Support Systems. The final note depicts the most common tool used to develop forms of ICW: 6) Authoring Systems.

1. **Stand-alone CBT Systems:** This type of CBT is designed to run on a stand-alone computer. (Note: It can also run on a networked computer, but the training is delivered via the computer that the student is using. This training cannot be transmitted over the network, and the computer cannot be actively communicating with the network

while the training is in session.) Typically, the training is distributed on a CD-ROM or, rarely, on a disk. Many companies offer "prepackaged" or "off-the-shelf" training modules on common topics such as customer service, dealing with diversity, project management, grade school basics, etc. In addition, many companies also offer customized training development services. Generally, only one user at a time may utilize the training package (unless you purchase multiple copies); however, a CD may be used repeatedly for different users. (Specifics depend on the licensing arrangements made with the training supplier. This will be discussed in more detail later.)

2. **Web-Based Training/On-Line Instruction (WBT):** This is the latest development in CBT. Training modules are located either on the Internet or on a company's internal intranet. WBT can be conducted while the student is on the net, or can be downloaded to the student's computer to be conducted off-line at a later time. WBT offers many advantages over traditional (stand-alone) CBT. For starters, WBT is more flexible than traditional CBT in that information can be quickly and easily updated and transmitted to all users simultaneously at a low cost. For stand-alone CBT systems, updating information can be costly

and time-consuming. WBT offers the student the ability to interface with the instructor and/or other students by asking questions, discussing issues, and receiving feedback. This adds a more "human touch" to the training. It would be virtually impossible to anticipate all the questions a student may have and incorporate them into a tutorial program. Another benefit is that a networked training system can automatically update the Training Manager/Administrator as to what training is being conducted. Also, students with home computers and modems could do the training from home, if desired.

3. Intelligent Tutoring Systems (ITS): ITS's are a more advanced version of the tutorials discussed earlier. The goal is for the computer/tutor to be able to assess the student's current level of knowledge, anticipate all of the student's needs, recognize the student's mistakes, and tailor the instruction to the particular student. Therefore, two students taking the same course may travel a different path to course completion. Ideally, ITS's would be extremely beneficial as they would offer the same expertise as a human teacher, but would add the individualization that is not possible in a classroom. Unfortunately, technology has not yet solved the problem of

making a computer "intelligent" enough to function as effectively as a human teacher. ITS's may provide exceptional training in the future, but currently they are very difficult and expensive to develop.

4. **"Just In Time" (JIT) Learning:** JIT learning is a concept that is not unique to CBT. It can also be applied to traditional training methods. However, CBT is particularly suited to JIT training because of its flexibility or ease of availability. JIT training "compresses the interval of time needed to implement change by bridging the gap between identification of a training need and its administration" (ASK International, 1996). More simply put, JIT training is conducted precisely at the time the employee needs it; there is no waiting for a large enough group of students to justify conducting the training. JIT training is particularly beneficial because it links the learning to the job and ensures a transfer of learning.

5. **Performance Support Systems (PSS):** PSS is similar to JIT training in that information is given to the employee precisely at the time it is needed. However, PSS has other functions besides training. A well-designed Performance Support System can "function as a reference librarian, an expert advisor, a patient tutor, and an administrative

assistant" (ASK International, 1996). Basically, a PSS automates/replaces the old concept of performance, procedure manuals, and job aids. The employee can look up information as it is needed to perform a particular job function, or go through tutorials as needed during "down time" to refresh his memory of policies and/or procedures.

> PSS is not just technology within the ICW sandbox. It is also objectives. Here our objective is to fully integrate support with the operational environment in a way that helps the system user do his or her job. It is what we used to call a 'force multiplier.' It is not usually considered entry level or basic training. PSS requires that the training element and the operational element be 'connected.' This is not true for ITS technology which only must connect to the sources of its artificial intelligence. JIT could go either way. (Moritz, 1996)

6. **Authoring Systems:** Authoring Systems are computer programs written to allow the user to design and implement their own ICW or CBT/WBT. When CBT was first implemented, it was necessary to be a computer programmer to be able to develop computerized courseware. With the introduction of authoring systems, it is now possible for the nonprogrammer to design courses. However, they are not foolproof; the course designer still needs to have a lot of computer knowledge to be able to use them. Therefore, companies that do not have an automation department as part of their staff

should consider using prepackaged software or a CBT course design consultant/firm that can customize a training program to their needs.

> You can go to a single source for finished ICW with nothing or you can go with a studied set of requirements. You can design your own curriculum, using individual courses from several vendors (assuming they are compatible with equipment and objective). Courses can be COTS or custom developed by a firm specializing in such work, or you can roll your own. In the last case, you will need the expertise to develop courseware on one or more authoring systems. In any case involving custom courseware, you will need to consider how the courseware will be maintained.
>
> It is the authoring system that gives your courseware the logical power to implement or fail to meet the requirements of the courseware based on intended technique, level of intelligence, interactivity, or other elements of pedagogy or technology. Limited power authoring systems yield limited capability courseware (i.e. The authoring system's capabilities make it possible to think of using PSS, ITS, JIT, or whatever other approach seems appropriate. (Moritz, 1996)

III. REVIEW OF CAPABILITIES AND REQUIREMENTS

This chapter reviews the capabilities, requirements, attributes and implementation considerations of authoring systems that must be evaluated in order to perform a complete cost/benefit analysis. Additionally, the current capabilities and requirements of the authoring system at FLEASWTRACEN are described. The information contained in this chapter was primarily extracted from interviews with ICW producers as well as its users.

A. INTRODUCTION

Comparing authoring tools requires training developers to identify their *target objectives, environment,* and *audience*. Unless these three are clearly identified, it's nearly impossible to evaluate the productivity ratios, cost-effectiveness, and other pros and cons of authoring tools.

Training objectives will determine how *complex* the lessons must be, and whether video, audio, animation, and certain other presentation tools are required. For example, one certainly can't present an effective music lesson without some means of producing sound. Suitable authoring tools will therefore include a means to handle audio files.

The **target computer environment** (operating systems,

stand-alone vs. networking, etc.) is critical. Training developers must consider whether existing computer systems will support the intended training, what changes might be accommodated by the budget, and a host of other factors.

The **target training population,** their experience levels and learning abilities largely define how *extensive* the CBT lesson and supporting material needs to be. On-line help, remediation training, and pace of the course are just three areas that must be tailored to the students for maximum training effectiveness. Additionally, the *size of the target audience* is important to perform a cost/benefit analysis per student.

B. DIFFERENCES IN AUTHORING CAPABILITIES

All authoring tools are not the same, and dividing them into categories for comparisons is a challenge. The majority offer the most commonly asked for utilities and also appeal to the largest customer base (the "K-Mart" approach to marketing). Others are streamlined to be inexpensive and simple for the low-end market (i.e. "CBT lite"). Still others offer unique features to a select and demanding clientele (like a "gourmet food" shop). Fortunately, there are ways to hold them up to the light and pierce the fog of marketing hyperbole.

All authoring tools are the product of marketing and technical compromises, and such design features as "icons vs. scripts." These, along with the operating system compatibility, will forever define the limits of that authoring tool.

Icons vs. scripts is a choice between simplicity and power in an authoring tool. For your lesson development, a vendor can either provide you with on-screen buttons and graphics to click your mouse on, or require you to type command lines for your lesson development. Think of it as the difference between using Windows and DOS.

Many of today's authoring products have opted to deliver icon-driven tools, and the average CBT developer prefers their initial simplicity of use. Projects can be started quickly, as little up-front training is required for their use, and little originality is required (or supported by the tool) in the development effort.

Note that a strong case can be made for use of scripting languages as well. Training tasks requiring a high degree of student interactivity or unusual originality, or developers with training objectives of greater than average complexity or whose finished product may be exported to a variety of operating environments, would all be better

served by a scripting language. In the former case, you are limited by the design constraints of the icons. In the latter, although no authoring system does everything, a good system will bring you to the limits of your imagination or the hardware capabilities -- whichever comes first.

One final consideration in the "scripts vs. icon" (i.e. power vs. simplicity) debate: it is entirely feasible to adapt the more powerful scripting tools to make them simpler to use, *but it is not practical* to start with a simple tool and make it more powerful.

Non-UNIX vs. UNIX compatibility rather neatly divides authoring tools into two broad regimes; underneath those two headings are various operating systems. The primary distinguishing feature of UNIX operating systems is their power and versatility. UNIX systems offer developers and programmers much greater control of the computing environment, though with a commensurate increase in the initial learning curve and in system configurating and administrating complexity. "Everything, it seems, in UNIX can be configured. The knowledge of how to do this is relatively scarce. Fortunately, it is hardly ever needed to run applications under UNIX". (Moritz, 1996)

UNIX authoring tools are invariably more capable than

non-UNIX, but are much less common. As less than 20% of all computers are UNIX systems, commercial vendors shy away from developing UNIX CBT tools. Instead, they overwhelmingly target the more familiar environments (DOS, Windows, Macintosh, etc.).

> One of the most important attributes of UNIX is its adherence to international standards of interoperability. UNIX is vendor neutral (unlike Microsoft Products). Every UNIX system, no matter from what manufacturer or vendor, is interoperable with every other UNIX system and with Windows and NT systems equipped with software that recognizes those standards. Microsoft has been trying to set its own proprietary standards. Every copy of Windows or NT comes from a single source, whereas each vendor maintains its own version of UNIX optimized for its own machinery or marketing ends, yet they all fully interoperate under the international standards of the open systems architecture. This is becoming significantly important in the Navy and DOD, especially in major cross-service organizations such as Military Intelligence. (Moritz, 1996)

Naturally, operating system compatibility has implications in the selection of authoring tools for CBT development. New projects require a lot of testing and redesign, and each transition between operating systems raises the chance of software incompatibilities.

While it is possible to develop training in one environment and offer it to students in another, it is generally not a good strategy. Computer-Assisted Self

Training (CAST) is the one notable exception. Among all the tools available, only CAST is well established and robust in migrating CBT between UNIX and non-UNIX platforms. ICON Author is now beginning to offer non-UNIX development and UNIX presentation, but only time may prove that a success.

Special features often distinguish authoring tools. Some offer extensive add-on utilities for the CBT developer, others are "bare bones" kits. The demands of your project should define your needs.

"While there is no standard approach for comparing products, evaluators should assess all products compatible with their operating environment which seem to offer at least 80% of (but not significantly more than) their desired functionality" (Carpenter, 1996, p.7). As when shopping for a new car, evaluators should prepare a "wish list" of features and a "will settle for" list.

To create that functionality "wish list", evaluators must be familiar with the CBT development process and their local training objectives, environment, and audience. This virtually eliminates relying upon contracting and acquisition offices as product evaluators.

Evaluators *are cautioned* not to be unduly swayed by extravagant features in an authoring tool. Utilities such

as automated lesson flowcharting, student registration data bases, and audio/video handling tools can either save CBT developers a lot of time and effort, or unnecessarily complicate a project and increase costs.

C. IMPLEMENTATION CONSIDERATIONS OF CBT AND ICW

Following are implementation considerations of both CBT and ICW. These considerations are created from a collection of magazine articles, Internet information, interviews with ICW developers and my own views based on my field research.

1. Resources Available: Many companies offer training software and/or course design consultant services. One method for finding consultants is to "surf the net." Each net search I tried yielded anywhere from 10,000 to more than a million selections (using the search term "computer-based training"). Other terms that could be used for a search are discussed later.

2. Course Selection: With the wide variety of companies and packages available, how does a Training Manager select which one would work best for his organization? ASK International (a software development company) recommends that you consider the following questions and compare the answers to your training objectives before purchasing "off-the-shelf" software:

a. Will it run on my system? If not, how much will it cost to upgrade my system?

b. Will it run on a network?

c. Is it easy to install?

d. Does the software make good use of the computer's capability, or is it just a taped lecture? What level of interactivity do we need for the training to be effective (and superior to a lower tech solution)?

e. Are there book marks?

f. Can the course be customized? (i.e. Does it allow you to add your company logo or customize any of the content?)

g. Does it have a library? (i.e. Can you store items you develop in a library for later use?)

h. Does it have a notepad that allows the student to take notes during the training?

I. Does the course appropriately evaluate learners? What is the company's culture regarding testing?

j. Can I track user information and test results?

k. Is the documentation complete, useful, and readable? (Three types of support documentation to look for: 1) the documentation that tells you how to install and use the program, 2) learner support -- workbook/job aids to

accompany the program, and 3) documentation that allows you to use the software in a classroom setting if desired.)

l. Is the program easy to use?

m. Is the course logically sequenced?

n. Is the content accurate?

3. Costs/Licensing Factors: As mentioned earlier, implementing CBT is not cheap! "A 1994 study of computer-based training found that the mean number of hours required to create a single hour of courseware was 228. At a conservative rate of $100 per hour, it works out to be more than $20,000 per hour of courseware." (Reinhardt, 1995) However, CBT is also an investment that pays big dividends over time. While it is true that the initial development costs are high, CBT has few recurring expenses (such as life-cycle management costs or the costs of updating the lesson material). Therefore, the more it is used, the lower the average cost per user. In fact, most CBT programs cost less per user than traditional training programs. Recall that traditional training programs often require travel expenses, instructor salaries, facilities rental, etc. which must be incurred for each individual sent to the training and/or for each time the training is conducted. d'Vinci Interactive Computer-Based Training estimates that total

costs for CBT (including development, delivery, and maintenance) over a five-year period result in a 28% savings over traditional classroom training. This figure does not even take into consideration the additional savings realized from less time lost off the job (lost productivity) due to the CBT taking less time than traditional training.

The cost of CBT depends on many factors including the type of program wanted (off-the self/prepackaged vs. fully-customized) and the number of users the buyer intends to use the program for (other factors also include the simplicity or complexity of the topic and the quality of the product). The formula that best depicts the total cost of an ICW product should be based on a "cost per recipient" basis.

```
   Acquisition cost of the software
+  Development costs of the lessons (salaries, etc.)
+  Maintenance cost of the software
+  Maintenance cost of the courseware (updates, etc.)
=  TOTAL COSTS
/  Total number of recipients
=    TOTAL COST/RECIPIENT
```

Note that these costs do not include the cost of the delivery platforms. If the existing equipment does not meet the training requirements, the cost of hardware upgrades are additional.

My research found training modules available for as low as $50 or as high as $800 for a single user. Each software

provider has its own licensing requirements and pricing options which can be very complex. Some offer "video-style" pricing (once you buy a copy, you may pass it around all you like, but not duplicate it), while others offer "software-style" pricing (each user is required to have an additional license and/or copy of the original CD.) Some other options available include:

a. Single user license -- for use by a single user on one workstation.

b. Multiple user license -- a master diskette with duplication rights up to the designated number of users.

c. Network license -- one network server with a designated maximum number of users and may be for a set time period (i.e., an annual license).

d. Corporate site license -- similar to the network license, but the price varies by number of users.

4. **Program Planning:** Planning for CBT should include the components similar to the elements used in planning for any other training program. The steps that deserve special attention are discussed below:

a. Establish a Basis for the Planning Process: Research shows that the biggest hurdle to implementing a CBT

program is convincing upper management that the benefits to be derived are worth the initial investment. Therefore, establishing support is a crucial element to being able to implement a Computer-Based Training Program. The trainer should save this step until after completion of a thorough needs analysis, researching budgetary issues, and calculating the long-term payoff of implementing CBT.

b. Identify Program Ideas/Sort and Prioritize Program Ideas: A thorough needs assessment is essential. Will implementation of a CBT program significantly improve the quality of training, or will a low-tech solution work just as well or better?

c. Develop Program Objectives: Do the objectives lend themselves to CBT? If yes, what type of CBT system (stand-alone CBT training modules (prepackaged, custom-designed, or a "roll your own" authoring system) or WBT) and what approach (JIT, PSS) would work best to accomplish the training objectives??

d. Determine Format, Schedules, and Staff Needs: Who will be involved in the research and implementation? Is there a "computer expert" on the staff who can assist? If you choose WBT, who will be responsible for keeping it up-to-date? For courseware, who will design the training

modules?

e. Prepare Budgets and Marketing Plans: Budget is a major consideration in the decision to implement CBT. Although the start-up costs may be high, it is important to consider the long-term budgets as well. Will the initial investment in CBT save your organization training dollars in the future?

f. Design Instructional Plans: What learning format will work best to meet the objectives (tutorials, drills, etc.)?

g. Coordinate Facilities and On-Site Events: How and where will the training be conducted (individual desk top vs. company "learning center")?

D. ATTRIBUTES OF AN IDEAL CBT AUTHORING TOOL

Regardless of the project, there are several attributes which make up an ideal authoring tool for the development of Computer-Based Training, especially for automated courseware which must be delivered to multiple or remote locations. These features are what I concluded to be the most important based on the results of my research. The numbers and standards noted throughout this section are a consolidation of the numbers provided to me through my interviews with CBT and ICW developers and users.

1. Proven Capabilities

It is generally unwise to risk a project's long term success on an untried product, yet far too often, project managers opt for the "latest and snazziest" product they find.

Only established users can judge how well a product performs over time, and where its strengths or weaknesses lie. Product comparisons are much more credible when other users of the product have been consulted for their insight.

2. Acceptable Degree of Complexity and Productivity

A development tool is only as good as the useful work you can derive from it. In the CBT arena, a product's potential utility can be estimated from its "productivity ratio". This ratio is obtained by dividing the number of *development hours* invested in generating courseware, and *presentation (or contact) hours,* which describes the amount of time required to present an automated lesson to a typical trainee. The development hours include curriculum development time, lesson flowcharting, graphics development, programming, and all other aspects of the production task.

At the top end of CBT development, the standard productivity ratios for highly interactive, graphically-supported, stand-alone courseware with complex learning

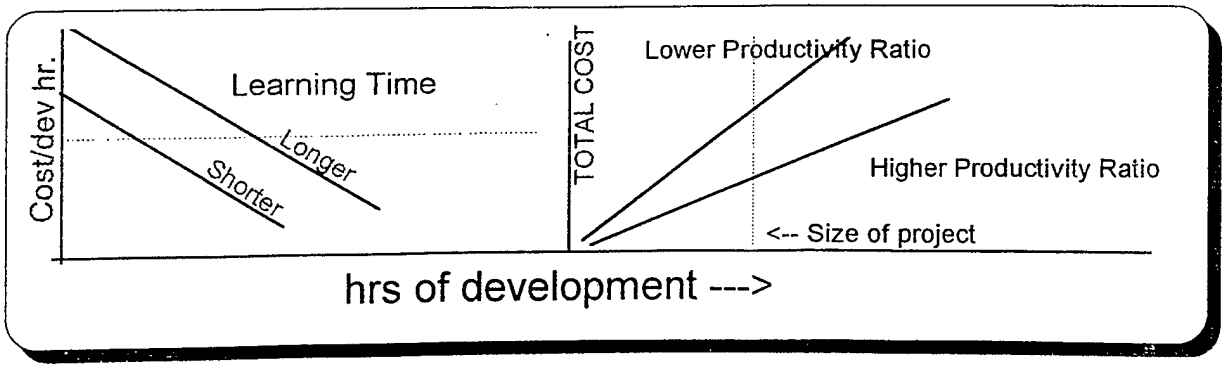
objectives, using sophisticated (scripting) authoring products, range between 200:1 to over 300:1 (depending on the level of interactivity and video technology desired). Many icon-based development tools may require somewhat less time for development, with a considerable sacrifice in capability and/or quality.

Another measure of a product's potential utility is the learning curve new users experience. As for complexity, one can expect a sophisticated scripting tool to require a two to six month learning curve for new users of the product, particularly those users inexperienced with CBT development in general and/or the operating system of their computer workstation. The learning curve is going to be significantly high for someone who is not real competent in or knowledgeable about their operating system, let alone CBT development in general. If a person is proficient with their operating system and has some understanding of CBT development, the learning curve is notably less. The reduced learning curve is most apparent when the author has mastered CBT development and is able to automate his routine authoring tasks. Most icon-based authoring tools require only two to six weeks to learn, as many routine authoring tasks are already automated (provided by the manufacturer),

and there is no direct interface with the operating system. These authoring tools require only comprehension of how to use a mouse and the ability to used a word processor.

Note that initial ease of use and productivity ratios *should not be* the determining factors in product selection, and evaluators must guard against overemphasizing them. Many quick to learn and easy to use tools may prove inadequate in the long run, while complex "scripting tools" can be made simpler to use, sharply reducing their learning curve and raising their productivity.

The costs associated with learning an authoring system are incurred once for each person who will be developing. It becomes less and less important as the amount of development increases. In the left graph (see below) we see that the cost/hour of development always drops below a given value. It just takes more hours of development to reach that point when you incur higher learning curve costs. This happens because the learning time cost is not proportional to the number of development hours.



Productivity ratios, on the other hand, become more and more important on the cost side of the analysis as you increase the amount of development to be done. The total project costs associated with development (not learning a system) are not fixed; they are proportional to the number of hours developed. For a given size of an effort, higher productivity ratios always lead to lower costs.

3. Versatility/Adaptability

Any product chosen as the tool for CBT development (especially exportable CBT) should be viewed as a candidate technical solution for other possible tasks. Because the initial investment in computers, software, developer training, and other resources are significant, it only makes good business sense for an agency to capitalize on that.

Another "cost-effectiveness" consideration is the prospect of using your product to train at multiple sites or for extended periods. This also depends upon an authoring product's versatility/adaptability, because the student environment varies widely between sites and over time. Tools adaptable to a variety of media (DAT, CD, etc.) and environments (stand-alone systems, client-server environments, etc.) minimize the risk of obsolescence.

Another important aspect of versatility is a product's

ability to interface with a variety of data formats (imagery, graphics, database, etc.) and network protocols (communication, storage, etc.). Developers may require access to many sources of data, pictures and sounds in order to produce effective training lessons.

4. Flexible Licensing and Reasonable Cost

CBT software is definitely not a "buy it once, pay for it once" proposition. Beyond initial purchase costs there are periodic licensing and support fees, requiring special budgeting plans at the local level, or annual submissions to the budget process.

Many project managers fail to realize that many vendors earn the bulk of their revenues after the sale. They find ways to charge customers for annual technical support, periodic software upgrades, or they charge by the number of computer workstations uploaded with their software, or number of courses built with their CBT tool. Some have even found a way to collect "royalties" by charging every time their authoring tool is used to present a lesson to a student.

Some vendors may sell licenses on a workstation-by-workstation basis, others by the number of developers involved. By far, the most cost-effective arrangement for

any large scale project is for "site" or "project" licensing. Essentially a "quantity discount", such agreements allow for full-scale development and broad student populations at a cost much less than station-by-station licensing.

5. The Product Stays Current

Through studying market trends and learning a little about an authoring tool's history, it's possible to predict whether the tool is likely to remain viable or grow obsolete. Much like market analysts, an evaluator can predict a product's growing popularity or demise by consulting users, trade journals, and vendor literature.

Profitability also impacts a product's future support. Vendors will continue upgrading a winning product, but cut their losses on unpopular ones by reducing their support staff, stretching out time lines between upgrades, or eliminating the product altogether.

Some products have an outstanding "track record", continually upgrading to meet new industry standards, offering drivers for the latest hardware devices, and breaking into new markets. Others fare poorly under scrutiny. Evaluators should pay particular attention to whether a vendor seems committed to staying up-to-date, not

by asking them (because they will fib), but by studying their history, their scheduling of new versions, etc.

E. OVERVIEW

Dozens of Computer-Based Training (CBT) authoring tools exist in the commercial marketplace. Finding the one that's best for your project is a matter of defining your requirements, understanding the broad capabilities and limitations of CBT tools, and recognizing which attributes are desirable in the candidate products.

F. CURRENT REQUIREMENTS AT FLEASWTRACEN

Based on the authoring tool requirements identified at the beginning of this chapter, this section outlines constraints and needs of FLEASWTRACEN. These requirements were identified by the FLEASWTRACEN ICW developers and myself based on information obtained through my research.

1. Training Objectives

The lessons being developed for the AN/SQQ-89(V) sonar trainer are fairly complex. They involve audio, graphics and animation, with an interest in incorporating full motion video, voice recognition, improved audio, touch screen and 3-D graphics.

2. Target Computer Environment

FLEASWTRACEN is currently utilizing an operating system

consisting of a twin screen (two 20" monitors, each with #9 video cards and 4 MB ram) system. The system is operating through Windows 3.1 for both the authors and the students. In the classroom, they utilize a mini Local Area Network (LAN) running Windows for Work Groups.

3. Target Training Population

The students attending these classes are a mixture of "fleet returnees" (students who have been in the Navy for a period of time and have served in the fleet), and personnel just out of "boot-camp" and have very little experience with *any* sonar system. They tend to be younger, energetic and anxious to learn. The sonar system they learn, no matter what rank they are, is new to each student. Due to the students being less familiar with the subject, the lessons need to be very extensive and self-paced, and include on-line help and remediation training.

4. Additional Requirements

In addition to the above requirements, the authoring package must meet the following demands:

a. Full Automation

One goal is to fully automate AN/SQQ-89(V) sonar training for unassisted presentation on computer workstations with the enhancements previously noted under

Training Objectives.

b. Globally-Distributed, But Centrally-Managed

The training will have to be globally-distributed to other training commands, ATG's, ships, etc., but centrally-managed (distributed and modified) at FLEASWTRACEN.

c. Employable On Available Hardware

It is desired that the training packages be employable on a wide variety of hardware platforms. This is in order to use the available hardware at the user's location and avoid the need to purchase additional training delivery hardware.

d. Courseware Must Be Classified

The courseware will be classified at various levels up to Top Secret Security Clearance Index.

e. Single Commercial-Off-The-Shelf Product

A single Commercial-Off-The-Shelf (COTS) product that is compatible with other COTS software programs (i.e. graphics, audio, etc.) is desired. A COTS product that is interoperable with the current graphics, audio, etc. packages is preferred in order to reduce the costs of changing authoring packages. However, it may be more cost-effective to purchase a package which contains the "add-on"

programs.

f. Flexible

It is desired that the authoring tool be flexible enough to handle the existing applications, data formats, or media, and function in the DOS, Windows, and UNIX environments.

g. "Intelligent Tutoring"

The product must offer "intelligent tutoring" (robust branching capabilities, an attribute of an authoring system) for cost and efficiency considerations.

h. Easily-Converted

It is mandatory that the current system be converted to the new system and installed easily. The conversion and installation costs are then associated with the costs of the project that were delineated earlier under the Implementation Considerations. The lessons that are already produced must also be easily converted to the new authoring software package in order to avoid excessive conversion costs that would make the purchase of a different authoring tool cost-prohibitive. If a candidate authoring system can't support easy conversion, the cost of conversion may be high enough to make that candidate cost-ineffective when compared to others. In addition, it must be user-

friendly to the authors, so that they may make it user-friendly to the students, in order to avoid severe start-up costs due to low author productivity levels.

G. CURRENT CAPABILITIES AT FLEASWTRACEN

The current system at FLEASWTRACEN allows the student to view each lesson sitting behind a computer work station which is composed of two monitors and a headphone set. The audio presents the lesson material in place of an instructor. The video display complements the audio with graphics, diagrams, and/or pictures. Each screen displays different items. For example, one monitor may show an outline of the material to be addressed in the lesson, while the other screen displays a picture of the particular system being discussed in the lesson.

FLEASWTRACEN currently uses Mandarin for Windows and Solo for Windows to develop their training courses for the AN/SQQ-89(V) sonar trainer. The present productivity ratio (total authoring time to produce one hour of a lesson in Mandarin) varies from 200:1 to 650:1. This variation depends on whether the lesson being developed is an "Exercise Lesson Model" or an "Integrated/Tutorial Lesson Model." Additionally, Navy personnel are afforded more time than contractors. The total authoring time consists of four

stages: 1) Development Stage, 2) Storyboard Stage, 3) Implementation Stage, and 4) Post Production Stage. A breakdown of the steps involved in each stage and the time limit for each step are depicted in Appendix A.

Mandarin offers the following capabilities. These capabilities were extracted from marketing material and information gathered on the Internet. It should be noted that Mandarin was considered by the Air Force in their evaluation discussed later in this paper (Carpenter, 1996). However, it did not meet their criteria for exportability across the web and flexibility from an authoring point of view. Although the price for Mandarin is very competitive, the Air Force evaluation team did not feel that Mandarin was progressive enough and would soon be technologically outdated.

1. Complete Design, Development and Documentation System

Mandarin for Windows supports the complete range of tasks associated with instructional Systems Design and the production of Interactive Courseware.

2. An Aid to Quality Control

With the above level of data on project and task control, Mandarin for Windows provides mechanisms for an

unprecedented level of quality assurance in the development of multimedia applications for training.

3. A Rapid Prototyping Environment

Mandarin for Windows encompasses the complete development process from the definition of training objectives through story-board specification to on-screen development of media and lesson structures.

4. Team Support Environment

Mandarin for Windows provides the first database driven development system. This allows multiple users engaged in different component tasks to work simultaneously on their individual tasks at their own capability level. This network database of information gives the manager maximum visibility on status and progress.

5. Increased Productivity

Productivity in the ICW development process has been the key issue since computers were first introduced for the enhancement of training. Mandarin addresses productivity from a total life-cycle perspective as well as the more prevalent authoring view. This not only saves development time, effort and expense but, by generating a completely documented system, guarantees minimum through life costs.

6. Object Oriented Design

A visual design tool leads to a fully documented object-oriented design which is easy to develop and maintain.

7. Life-Cycle Support

Mandarin for Windows takes the project seamlessly from the initial outline design, through detailed specification, setting of project standards, implementation, and integration into service.

8. Multimedia Presentation

All presentation objects are handled by a single command which supports text, Windows Metafiles, Windows Bitmaps, Video, Audio (Windows Wave) and Multimedia Movies.

9. Graphics

Current Installation range from 16 color 640 x 480 VGA to a twin screen system at 256 color, 1280 x 1024.

10. Extendibility

The system is backed up by a powerful authoring language which provides a comprehensive simulation capability. Additionally, preprogrammed, user-defined functions can be accessed through Dynamic Link Libraries.

11. Team Support

Mandarin supports a team-based approach to courseware

production. Enforces access rights to courseware objects, provides the tools to control the objects through an ongoing review process. Team operation is through a shared file server. Single user operation is also supported.

12. Standard Adaptive Lesson Control

A powerful student control panel is provided along with all the necessary runtime support so that the developer only has to develop the training material, not the training framework. The control panel can be adapted to particular needs.

13. Hardware

Author: The Windows minimum is sufficient but the recommended system is 486-25 MHZ, 8 MB RAM. Student: The Windows minimum is sufficient.

H. DEVELOPMENT OF A CRITERIA LIST

Utilizing commercial practices, technical materials and interviews, I developed a list of questions that I will use as my selection criteria when analyzing the competing companies of authoring tools. Additionally, I have employed the previously mentioned attributes of an ideal CBT authoring tool, implementation considerations, and the current requirements and capabilities at FLEASWTRACEN.

These criteria will be discussed in greater detail in

Chapter V.

1. Is the product a stable, enduring COTS product?

2. Is the software product "standards-based"?

3. Does the product offer broad functionality?

4. Is the product's return on investment high or low? What is its expected productivity ratio for courseware development?

5. Can the authoring tool be used as an "intelligent tutor"?

6. Is the product "progressive"?

7. Is the product suited to the target training environment?

8. Will it work on a network, supporting client-server architectures?

9. Is it fully featured? If not, is it adaptable?

10. Will the product satisfy the government's stringent security demands?

11. Is the product licensing relatively flexible and cost-effective?

IV. COMPARISON OF COMPETING COMPANIES

This chapter identifies and extensively compares seven competing companies that produce authoring tools.

A. SYNOPSIS OF "AUTHORING SOFTWARE" COMPARISONS

Over a period of seven months in 1992-93, the Air Force conducted comparisons of numerous commercial and government training development software products, for the conversion of intelligence skills training to an automated format. Product research was conducted by the Integration and National Systems office of the Air Force Intelligence Training Center, at Goodfellow Air Force Base (AFB) in Texas. A team fully-knowledgeable of the scope and dynamics of the intelligence training mission, its technologies and software applications, and its future direction conducted the research effort. (Carpenter, 1996, p.10)

A brief discussion of the product comparisons found through their research, as well as the information I uncovered in my research, follows. This paper makes no attempt to offer elaborate and unnecessary detail, as the selection process and criteria have been more thoroughly described previously.

The selection process was complicated because the

training objectives and delivery approaches are constantly changing, influenced by new technologies, government policies, DOD alliances, variable resources, and shifting geo-political threats.

B. PRODUCTS COMPARED

The Air Force reviewed sixty-six (66) product vendors. The principle products assessed in their selection process included (in no particular order of precedence):

ASK-ME 2000 & ASK-ME Professional
PHOENIX (Standard, LAN, & Micro)
Authorware Professional
Tutor-Tech Hypermedia System
Interactive Virtual Video (IVV)
Creative Course Writer (CCW)
Ridgewood-Authoring System
SOCRATIC Authoring System
Desktop Training System
Training & Education Language (TEL)
D/Vision & D/Vision Pro
Hyperwrite Professional
Syllabus for Windows
CSR Trainer 4000
Icon Author
Quest Multi-Media
GAIN Momentum
Teachers Aide
Tool Kit
Course Plus
Lesson Card
CLAS-CBT
Course Builder
Amiga Vision
Unison
Summit
Foundation
X.TIE
Accord
Merlin
TIE
CAST
IMSATT
AIS-II
Act III
SABER
Quatrain
Natal
SAM
IV-D

My research was much narrower in scope. I evaluated three of the top five vendors of authoring systems; Allen Communication, Inc. *Quest;* Macromedia's *Authorware;* and AimTech Corporation's *IconAuthor* (Conversations, 1996). I chose these three because they have been in existence for approximately 20 years and was therefore able to thoroughly research them. I also studied Sybase's *Gain;* Master Class Corporation's (MC2) *CAST;* Josten's Learning Corporation and LearnStar Corporation. I considered Computer-Assisted Self

Training (CAST) since it was the Air Force's choice of software. I examined Gain because it was a strong consideration of the Air Force due to its flexibility and network capabilities. Josten's Learning Corporation and LearnStar Corporation were researched, on the request of the Commanding Officer, FLEASWTRACEN, based on their extensive experience with educational institutions (grades K-12).

C. COMPARATIVE NOTES

Most vendors did not want to provide me with much assistance since I was not a procurement officer. However, I was able to extract information from various sites on the Internet. I used as many commentaries from users of the different software products as I could. (Note: Some of the comments are two years old and may be out-of-date at this time.)

When evaluating these products, the selection criteria previously outlined were used in order to ensure that FLEASWTRACEN's requirements were met. These notes serve as illustrative examples of the comparison process. As such, they should not be construed to be more than representative comments of the Air Force team's or my assessments of the individual products.

1. GAIN Momentum (representative comments)

Gain Momentum is a multimedia development system that operates in a multi-user, networked environment, utilizing Open Windows under UNIX. This authoring tool is designed for stand-alone applications or ones working with databases. The applications can be developed using the visual programming environment, Graphical User Interface (GUI). This environment is written in Gain Extension Language (GEL), a scripting language. This program features a time-line editor, object browser, user-defined classes and libraries, a binary port for receiving external data, and an enhanced SQL development environment. The system requirements are 100MB hard disk with 32MB RAM.

"Gain works well with UNIX and PC (Windows NT only) platforms, but has problems with full functionality on Macintoshes. The tool works by designing the interface with an interface builder and then adding functionality to the widgets with the built-in GEL scripting language. The GUI builder is fully functional, though it could use a slightly better interface. It uses a canvas and icon-placement of widgets, which are fully functional immediately so there is no compiling to slow things down." (Sybase, 1994)

Despite vendor rhetoric, Gain Momentum was

> too focused on *emulation of the systems and applications to be trained,* rather than providing a dynamic training interface (or a software bridge) to *actual operational applications* on the net. Ideally, a CBT development tool will not restrict itself to simple simulation of a target training application. Access to the real data bases and systems used by the trainee on the job offers the maximum level of realism in a teaching environment, and avoids the costs inherent in developing and continually updating emulations of target systems.
>
> Gain did not support operations on DEC equipment, which is a primary platform for intelligence processes, and is the first platform offering 64-bit technology at the workstation level. In reviewing their future objectives, Gain expressed strong doubts about the need to 'port' their product to DEC platforms. Furthermore, Gain did not appear to support (offer interfaces with) touch screens, laser disk players, and other media then pertinent to the intelligence training environment. (Carpenter, 1996, p.3)

Though overall, Gain may be a strong product for some existing training arenas, the Sybase Gain vendors seemed to repeatedly demonstrate a lack of insight and strategic direction in addressing the future needs of training systems and applications. This was evidenced in several ways:

a. Gain's disinterest in porting to 64-bit platforms, when clearly that technology was essential for effective use of virtual reality training, collaborative environments, rapid audio and video response, and direct access to large data bases (over 4Gb). Though this

criticality was emphasized to the vendor, by the Air Force, Sybase disputed its importance, and had no plans to migrate.

b. When Sybase, Inc. acquired Gain, they seemed uncertain what to do with it. Other Gain users were growing apprehensive, and trade journals weren't offering much reassurance. The Air Force's "apprehensions seemed validated in Feb 95, when Info World reported SYBASE planned to halt development *of Build Momentum* and *Enterprise Momentum* tools in favor of using *Powersoft's Power Builder* tool, and would be removing roughly a hundred employees from GAIN development projects" (Carpenter, 1996, p.3). Triton Services has recently acquired the license to produce Gain Momentum. It was so recent that I was unable to obtain any information, as to future improvements, from Triton.

The pricing for Gain Momentum starts at $10,000 for single-developer licenses. "Gain was much too expensive for a Department of Defense acquisition, and their licensing wasn't flexible enough to allow for compensatory cost savings. Even its potential to save the Air Force millions failed to justify its high investment costs." (Carpenter, 1996, p.2) Additionally, "Gain is a highly complex package, with tons and tons of capability. Unfortunately, this wealth requires tons and tons of code. Gain has frequently

been called a resource "hog" because of its huge size on disk. Some reports have also found it slow (it has lots of overhead, apparently)." (Moritz, 1996)

2. ICON Author (representative comments)

Icon Author has recently been designated as the Navy Chief of Naval Education and Training (CNET) training standard. Icon Author is a cross-platform multimedia authoring tool that has an icon-based development environment with no scripting language. It allows developers to either embed content or to keep content and structure separate. Icon Author features Internet accessibility and operates on OS/2, Windows, Windows NT, UNIX, and Macintosh (in October, 1996) platforms. It requires a minimum development system consisting of a 386 PC AT, 4MB RAM, 11MB disk space and a 4-bit (16 color) VGA graphics card. However, a 486 with 8MB RAM and 20MB disc space is recommended.

Like most authoring tools offering automated "story-boarding", ICON Author offered students *virtually no capability to branch out beyond their active CBT lesson.* Every nuance of a training presentation, every possible need of the student, had to be foreseen, story-boarded, and built into the CBT presentation. "In teaching skills which

require analytical thought (such as military intelligence), or many performance-based skills, ICON Author's 'locked in' presentation approach has historically proven less effective than allowing trainees to 'set their own course' (Carpenter, 1996, p.4).

A scrollable set of 60 icons are included in the package, with an ability to add customizable icons and expand the "Icon Library." By developing this library, an author is able to scroll through and find an icon which is used repeatedly. However, this may be time-consuming since it takes some effort to develop the icons and when scrolling, only six icons are displayed at one time.

This is not suitable if you needed to allow students to spontaneously query and access external applications, to move around more freely within or *move beyond* the inflexible story boarded lesson. This enables trainees to access on-line remedial training, to look for ways to better understand the teaching objectives, or practice new skills. That means that the training can be "student-tailored", according to the needs of each trainee.

Icon Author offers cross-platform specific icon libraries and file format compatibility between numerous operating systems. Applications can be moved between

platforms without a conversion process. This allows a single development effort and a single set of development skills to be leveraged across many platforms, saving significant development time and money. (Gery, 1995) However, after talking with the Air Force and others who reviewed Icon Author, I discovered that Icon Author has had problems with portability.

Icon Author's many features (including their technical support) are impressive, yet they are not all-inclusive. Users are limited to built-in tools and could not get deeply into scripting, where the real flexibility and power of an authoring product lies. Limitations also exist when information needed to be linked for access on a LAN. Development is interrupted when an author attempts to link to external editors such as graphics programs, text editors, calculator, and clip board. Development relies primarily on scrolling to find the appropriate icon, dragging it into the flowline and completing dialog boxes. It also does not offer a graphics editor; instead it links to Paintbrush.

> The Smart Object Editor is an external editor with little in common with the design interface of the main Icon Author environment. Because of this, users need to learn multiple development environments in order to create in Icon Author. Additionally, only certain kinds of objects can be designated as Smart Objects, limiting the type of

> interactivity that can be achieved with Smart Objects within Icon Author. Because of the detailed nature of assembling Icon Author applications, the learning curve can take a little more time than that of other authoring systems. Once familiar, the detailed nature can continue to reduce the development speed unless the developer becomes acquainted with and uses customized libraries containing custom composite icons. (Magel)

The cost of Icon Author varies significantly depending on the type of system and what it is used for. For a PC, the price is $4995; UNIX, $5995; Mac player $1495. However, if you can qualify for an education special, the price is $995. -Unlimited technical support is free for 90 days. Additional technical support costs $500 (standard) or $795 (premium) annually. If additional training above and beyond their five manuals is required, AimTech Corporation offers a four-day basic and a four-day advanced training course at an on-site price of $1500 per student. In addition, there is a $100 per copy distribution fee for UNIX users. For commercial applications, AimTech requires an additional royalty be paid for the run-time license.

"Licensing costs were always in dispute. The project was to benefit from reduced licensing fees, then it was discovered this was a conditional price, based upon a "handshake" and dependent upon a larger (unrelated) Air

Force buy. Later, an educational discount was to apply, then was rescinded based upon the scale of the project in combination with other (unrelated) projects." (Carpenter, 1996, p.4)

"Icon Author was too restrictive in its interface with media for reliable operation in an exportable environment. Media drivers were 'fickle', in some cases requiring advance knowledge of a student's training workstation down to the level of their computer components' make and model. Such limitations quickly turn 'exportable training' projects into a global configuration management process." (Carpenter, 1996, p.3)

3. QUEST (representative comments)

Quest is a fully-integrated authoring system and is designed for developers at all levels. Developers work from "what-you-see-is-what-you-get" (WYSIWYG) displays and floating toolbars to assemble graphics, text, audio, video motion, buttons and animations, or to set up branching and interactions. Users work with a highly-intuitive, object-oriented interface to directly manipulate and edit screen elements. The minimum system requirements for authoring include a 486DX 33Mhz processor with 8MB RAM, 3.1 Windows or higher, and a VGA graphics card. However, a 486DX 66MhZ

processor with 16MB RAM, and a SVGA graphics card is recommended. The minimum run-time requires a 486SX 33Mhz processor with 4MB RAM, 3.1 Windows or higher and a VGA graphics card. The space requirements for a run-time player is 1.4MB, for Quest programs it is 10MB, and for optional programs it is 25MB.

Quest's intuitive interface gives developers the flexibility to author at two different levels; the Design or Frame levels. At the Design Level, instructional designers can make big-picture decisions about overall structure and interactions. Working with "postage-stamp" or "thumbnail" representations of actual modules, lessons, and frames, developers have an immediate view of how the title content looks and how it links together. This not only encourages sound design, but reduces the mistakes that are made when developers are forced to work at a micro level too soon. It allows the users to lay out the structure of the course independent of content in "big picture" perspective. Design Level relies primarily on icon ribbon tools and scrollable "Quick Frame" libraries, which are structures of frames with branching logic already saved.

Artists and content experts can work inside the actual frame at the Frame Level, viewing the screens exactly as the

end-user will see them (thus the WYSIWYG environment). All functions of the system can be accessed without exiting to outside editors. Graphics, audio/video options, controls, interactions, branching, and animations are all built into the system. In this mode, the author is able to create/import content objects using floating tool boxes. Media groups and functions are accessed through tab indexed card file-like toolbox with each "card" presenting a specific group of icons to be dragged onto the WYSIWYG frame. Quest utilizes a C-based language. This C script box is automatically assembled and may be amended or edited directly using basic word-processing actions. Quest supports all videodisc players that supply an MCI driver.

Quest Net+ (the latest version) is expanding into the Internet connection. This allows the user to update and add dynamic content to titles during run-time. Quest titles are able to read files stored on any web site and makes the most of current Internet capabilities for communication and data exchange within current bandwidth limitations. However, I could not find any indication of this software being portable across to other operating systems outside of Windows.

Quest was typical of the many DOS-based icon-

> driven authoring tools. It offered no value added except initial ease of use (only one of many criteria for selection). Like other icon managed software, QUEST only did what the vendor had the foresight and resources to include in their icon menu, and could not be readily adapted by the user to perform other needed functions. Often, developers discover such "easy" tools more difficult to work with, when they are forced to invent workarounds to overcome the limited functionality of the product. Training could only be developed and presented on common office machines. QUEST did not offer portability across operating systems, could not be globally distributed yet centrally managed, and proved weak on "intelligent tutoring" functionality. (Carpenter, 1996, p.4)

Quest 5.1 lists for $3995 for each concurrent user, with upgrades available for $500. Allen Communication offers unlimited technical support free for 90 days, via an 800 number. Additional technical support costs $500 (standard) annually. A five-day essentials and a five-day advanced training session is offered for $1500 per student (plus expenses with a minimum of four students for on-site training). Consulting services are available as well as full courseware development services.

4. Authorware (representative comments)

Authorware 3.0 is an icon-based authoring environment designed for people who want to create highly-engaging, interactive applications without having to learn challenging programming or scripting languages. It features point and

click hypermedia, powerful text features, seamless integration of fully-interactive Director movies, custom buttons, and two-way portability between Windows and Macintosh. The interface includes pull-down menus with a Data menu for calling functions and variables, a Library menu for linking to template code and content, an Attributes menu to control the appearance of visible objects on the screen, and a Text menu for controlling the objects. The minimum development system recommended requires a 486/33Mhz with 8MB RAM, 4-bit (16 color) VGA graphics card, and a 40MB hard disc, running Windows 3.1.

Authorware only provides a fixed set of 15 icons in the toolbox. The icon flow windows (composite icons expanded to a "next level") are non-scrollable and therefore cannot be longer than roughly a dozen icons before grouping is required. This set of 15 icons can accelerate the immediate authoring process, but it also puts more weight on the developer's skill at naming icons so that they can be easily recognized without have to open up each icon to reveal its contents. This can slow development down considerably as courses become larger and larger.

The learning curve is fairly fast on the level of assembling flow lines of icons and then placing content

within display frames for presentation purposes. However, assembling training applications with student record keeping features requires some sophistication and familiarity with the use of variables and functions.

The primary drawbacks of Authorware are the limitations in graphics capabilities (such as limited transitional effects and animation features), the integration of external executables (they are not readily apparent when authoring), and the limited portability between Windows and Macintosh only. Macromedia has developed a variety of software tools including authoring tools, 3-D modeling and animation tools, audio editing tools and presentation packages. Unfortunately, each tool is sold separately and each interface is unique, so developers have to buy and master many packages to reach full productivity.

The cost of Authorware is $4995 with unlimited technical support for 90 days *beginning with the first call.* After the initial first call, priority access technical support, in addition to free technical support, is available via online services (an 800 number). Additional technical support costs run $699 to $949, annually. Supplementary training is available throughout the U.S., but a price for this service was not available.

5. Computer-Assisted Self Training (representative comments)

Computer-Assisted Self Training (CAST) is a powerful authoring system able to accommodate even the most demanding training requirements (although in some cases, it may require the assistance of other programs). CAST courseware is portable to a wide variety of character-based and X Window System terminals under Sun operating systems/GUIs, as well as other UNIX and non-UNIX environments. It includes facilities for courseware production, courseware presentation (with courseware integrity control), and computer-managed instruction (CMI). Master Class Corporation (MC2) is also very reputable for their expeditious technical support assistance.

CAST is oriented to the production of highly-interactive courseware, and is unique in its ability to invoke and control other applications resident on the system/network. By incorporating existing COTS and custom software into a course, the author may cost-effectively concentrate more on interaction with the student rather than on expensive simulation. With CAST, users have unlimited branching, computational facilities, and a powerful ability to evaluate an individual trainee's response/status concurrently with presentation of the courseware. CAST

features display and manipulation of multiple text fonts, graphics, imagery, and icons; custom-designed buttons and pull-downs/pop-ups; and video.

CAST is developed in the C language. However, the author does not see the C language. The scripting is done in the CAST Very High Level Language ((VHLL) -- a term the industry uses to describe the most recent high programmer efficiency and powerful languages emerging on the market). It has virtually unlimited portability (seemingly the highest in the industry, from my research). Some of the items encountered that limit portability, and no one in the industry has yet gained control over, are: the resolution of display, the depth of the colormap (number of bits used to specify color on a pixel-by-pixel basis), the actual fonts available, and the number of mouse buttons. "If you develop graphics for a true color display (24 bit depth) you cannot show them on an 8 bit deep pseudocolor display. So the developer must still consider the characteristics of ALL machines on which the courseware is to be presented, and either shoot for the common denominator or use CAST's ability to recognize the environment and branch to an applicable variant of a presentation." (Moritz, 1996) This almost limitless portability enables an author to transfer

and deliver courseware between graphic workstation systems using UNIX with little or no change, and a wealth of terminals and/or various client server configurations; including MS-DOS and Windows-based systems. CAST is also able to interpret student responses to questions and/or cues, allowing students to be more expressive in their answers vice yes or no answers.

By using a scripting language, the learning curve *initially* is greater. However, the language structure is much like English and is in the form of simple logic statements. This learning curve is quickly reduced as the authors master the scripting. Once the language is mastered, the capabilities of this software are virtually limitless (bounded only by one's imagination).

Master Class Corporation is continually looking for ways to improve their product. To date, they have implemented full motion video, touch-screen, 3-D graphics, closer view or "progressive disclosure" capabilities, and the ability to modify on-the-fly. While CAST has not yet implemented voice recognition, it uses external software and hardware components provided with the voice recognition system rather than doing the job internally, enabling you to use the "best" voice recognition system. Although not yet

fully developed, CASTs internal architecture is inherently multilingual. CAST will support French, Spanish, Greek and Cyrillic character sets. They are also researching enhanced graphics and sound, as well as incorporating animation with playback features into their package.

The pricing of CAST is also very flexible, depending on the number of components you want to purchase, and the number of trainees, operating systems, and developers who will be using the system. The price varies from $200 for a "single seat" license to an enterprise license which can cost up to $1 million. A customer is able to purchase a LAN, site, project, or enterprise license for one set price, depending on the amount of use that is required. Another option for the customer who is unsure about the product or its amount of use may purchase the product by "piece meal." The cost of the Producer (the development environment including the author's CMI (administrative scoring capabilities)) varies by the number of installations (not users). The Publication environment (a CREW license enabling the author to create script, protect changes from the student, view controls, etc.) costs approximately $4000 to $5000. Producer, Crew, and the Software License Manager (SLM) each have catalogue prices. MC2 allows a 50% premium

on CMI components. When you add CMI functionality to Producer or the SLM, you add 50% to the price of whatever Producer and SLM products you have ordered. Technical support costs also vary, and are proportional to the size of the order. Customer support is separately priced and available by the hour or other agreed-upon basis.

6. Josten's Learning Corporation / LearnStar Corporation (representative comments)

Both Josten's Learning Corporation and LearnStar Corporation provide Multimedia Interactive Learning Systems for grades K-12 classroom environments. Their software is preprogrammed (proprietary), with very little modification abilities. The authoring is done primarily in-house before it is delivered to the schools. The teachers are able to make some minor adjustments to the core curriculum programmed, but changes are very limited.

The software provides colorful CD-ROM graphics and full-motion video classes for more than 1,000 curriculum-based competitions. The students are able to compete against not only their own classmates, but also other students nation-wide who are linked through the same network. Both of these utilize a PC as their base units, creating a stand-alone or network system. Both systems are very user-friendly for the teacher and the students as well.

While these products provide live satellite competitions and produce excitement in the classroom, the competition may hinder some of the students' learning. The students that feel embarrassed or are shy will be apprehensive about answering the questions because their name, answer, and score are displayed on the television for everyone to view.

a. Josten's Learning Corporation

Josten's Learning Corporation supports the Apple, DOS, and Windows environments. The standard requirements for hardware with a Windows-based system is a 486SX-25Mhz with 8MB RAM (16MB swap file), 3.5" with 1.44MB floppy drive, 80MB local hard drive, a mouse, VGA or SVGA graphics 640x480x256 display and adapter, headphones, and a MS-DOS 6.22 and Windows 3.1 (or higher) operating system. Each student will be required to have access to a computer (unless they work in groups) in order to participate in the training.

At this time, Josten's is not interested in developing interactive courseware for the military.

b. LearnStar Corporation

LearnStar Corporation utilizes a pentium-based PC with 16MB RAM and 850MB hard drive with a quad speed CD-ROM,

sound blaster 16 stereo sound card, PCI bus graphics accelerator card with simultaneous SVGA and NTSC output, 14,400 BPS modem, trackball and 101-key keyboard, and a 14" VGA monitor. The software is loaded into this base unit and the lesson is sent to a television for display in the classroom(s). Each student (or group of students) then provide their feedback through Interactive Wireless Keypad Systems, which are powered by rechargeable batteries.

LearnStar is also operating in the Sports Bar arena. They offer their system for the customers to participate in nation-wide trivia tests.

The LearnStar product may be more cost-effective than the previously mentioned products for courses that are continual and do not change. Once the courseware is developed, it can be used repetitively, constantly decreasing the cost per student. This courseware would be best suited for "boot-camp" or "A" school students to teach the military rank structure, etc. It would provide a competitive learning environment which would require very little instructor involvement.

V. EVALUATION AND RESULTS OF ANALYSIS

The eleven criteria that I used to evaluate the competing companies and their products, and ultimately select CAST as the most effective and efficient ICW product for FLEASWTRACEN, are discussed in great detail in this chapter. I also outlined some productivity considerations that need to be studied when choosing an appropriate authoring tool. This chapter is concluded with additional benefits the Air Force discovered which ultimately led them to their selection of CAST.

A. SELECTION CRITERIA

Guided by commercial practices, technical materials, and interviews, I used the following criteria (as outlined earlier) in the selection process.

1. Is the product a stable, enduring COTS product?

I looked to the past for products with a good track record. I focused on those with an established user base in order to find *mature products with proven value*. That told me about their utility and reliability as well as the responsiveness of the vendor to the customer needs.

I looked to the future for an authoring tool which would serve FLEASWTRACEN for many years. The product would

have to remain available and technically supported through 2010. It was therefore essential to identify *dependable and forward looking vendors.*

In looking for future viability, I held discussions with other product users, visited some vendor facilities, tested their staff on their understanding of our needs versus their product, and considered such strategic factors as vendor standing in the marketplace and solid business practices.

Computer-Assisted Self Training (CAST) more than qualified. The vendor (Master Class Corporation) had established themselves as a commercially-viable company with a history of adapting their product to new technologies and to meet evolving user requirements. CAST had been in use for over nine years, with a number of established users (including the military intelligence community).

As the sole developer, Master Class held full rights to the product, and I could find no evidence of customer dissatisfaction, incompetence on the staff, or poor business practices. In short, there are no issues of any legal, commercial, or managerial nature which may jeopardize either the company or their product's availability.

2. Is the software product "standards-based"?

Software standards, both industry and government, have always been critical. *As a practical consideration,* a standards-based product offers assurance that it can be used with available hardware, and integrated with other applications. Most standards-based products are also broadly-marketed, and their quality and cost-effectiveness are influenced by marketplace competition.

There are also stringent government guidelines on software standards which affect both software acquisition, and software use within the Federal and DOD communities. These were a very real concern, for many products appeared not to meet those twin challenges. Any controversy over those (frustratingly conflicting) guidelines, or any shortcoming in meeting them, would likely come back to haunt or terminate an ongoing project while a more suitable product could be found.

During the review and follow-up licensing activity, I exercised particular care to ensure CAST met guidelines of the Federal government, Federal Acquisition Regulations and the Defense Federal Acquisition Regulations. I also verified this through the Air Force. "CAST established itself as compliant with all requirements" (Carpenter, 1996,

p.3).

3. Does the product offer broad functionality? Is it limited to training only certain applications?

Although the military community seems to be migrating to common software tools wherever possible, there still exist perhaps 40 to 80 computer applications which will require some degree of computer-assisted training in the training arena.

The Navy has insufficient time, money, and manpower to develop that amount of training with multiple authoring systems. I therefore sought an extremely versatile, *common product* which could automate the training of all of them. Again, CAST met that challenge. With its numerous utilities and unequaled portability, it appears CAST can be used to train any application, in any media that I have identified, and utilize almost any computer operating system required. It does not run on Macintosh's. It will only operate on NT boxes if it has a commercially-available X-server. In documentation and demonstration, CAST showed itself virtually platform and application independent.

Not only was CAST able to function on a variety of platforms, *it could also port a training package between operating systems.* I learned of one instance wherein a

vendor developed a training lesson with CAST on a UNIX-based SUN station and presented it without modification on DOS-based office machines. Previously, that was thought to be impossible. Since that time, the Air Force has developed CAST training on DEC platforms, and presented it on SUN and HP machines.

4. Is the product's return on investment high or low? What is its expected productivity ratio for courseware development?

Investment return is as much a concern to the DOD as it is to commercial firms. Our budget and manpower determine what resources we can invest in a project, and the tools we use either limit or enhance our productivity and training delivery.

The key determinants of an authoring tool's return on investment are: (1) legal limitations placed on use of the product, (2) size of the student population, and (3) the user's productivity ratio.

a. Legal Limitations

While many companies will make small accommodations in their licensing provisions, I discovered Master Class routinely offered licensing tailored to meet the user's specific project needs. CAST users acquire the licensing for the combination of software modules, operating

systems, and project size which best fit their requirements and budget. Goodfellow AFB discovered it could acquire a project license to develop all the training packages desired for all the platforms required.

b. Student Population Size

The ongoing Navy project (in order to meet the increasing technological advances) is to provide "exportable" training. The CAST license acquired by the Air Force (an enterprise license) is virtually non-limiting, allowing Goodfellow AFB to deliver fully automated training to 1000 concurrent users around the globe.

c. User's Productivity Ratio

A product's "productivity ratio" is a comparison of the number of hours of development time required to produce each hour of CBT for the typical student. Industry standards for high-end Computer-Based Training can exceed 250:1.

CAST offers initially moderate, then rapidly improving productivity, to achieve exceptional overall results. In phone interviews, users claimed initial development ratios ran between 250:1 and 200:1. After their initial learning curve, all of them improved this figure significantly, with many producing at under 100:1 (better

than twice the industry's target ratio), and *in several cases CAST was as low as 40:1*! This is achieved through three methods: a) *by enabling users to develop templates* for their "routine functions", which promotes efficient software re-use; b) by use of *keystroke capture*, which allows even non-programming casual users to become productive; and c) by using the helpful utilities included in the CAST development environment.

5. Can the authoring tool be used as an "intelligent tutor"?

A major resource challenge for automated training has always been providing computer access time to students. Minimizing each student's time on a training station translates directly into valuable machine time for other users. The opposing challenge is to ensure no student is slighted, in that the training presentation is adequate for each individual student's learning ability.

To address this, a product should support "intelligent linking" between various portions of the automated courseware, and with other applications on the network. This allows competent trainees to skip ahead in their courses, and provides the more challenged students with access to additional training materials to complement the

curriculum.

Several products offered such "navigation" capabilities, but most were bound within the active portion of a presentation. Others offered some degree of dynamic linking with other data bases and applications on the network, but were crude interfaces with limited adaptability. *CAST was the notable exception,* and made possible student access to the entire network. The only limitations on access were those intentionally imposed by courseware developers for reasons of appropriateness, security, etc. If need be, a student can start a CAST lesson, exit the ongoing training to access another lesson or application, then return to the original lesson to continue training.

In practice, this means students can train a little, switch to another application to practice their skills or do other time-critical work, and return to training when time permits. No system time is wasted, as 100% of the available system time can be shared between students and other users, with minimal interference.

6. Is the product "progressive"? Does the vendor keep it current with new applications and technologies, or let it get outdated?

Master Class Corporation continually enhances the

product. Each year, they invest the majority of their product income in expanding the utility of CAST (72% in 1994, more in 1995). Just in the past two years, this forward-thinking company introduced five significant enhancements, and is beginning work on yet another:

a. The new *CXC interface* was introduced to make CAST as versatile and application-independent as possible, allowing it to monitor, train, and control even an "unfriendly" application. It lets CAST intercept commands generated by an application (even one which does not provide data across standard processing lines), trap those commands, and then modify, emulate, or allow routine function of the command.

b. CAST was the first training product ported to *the 64-bit operating environment*. As early as March '93, it was used to develop training on DEC Alpha workstations.

c. Master Class recently introduced utilities called *"agents"* to allow users to specify the interface between CAST and target applications. These agents tell CAST what kind of messages to expect from applications, and in what format CAST should respond. Essentially, the agents translate the messages passing between CAST and the target application, engendering a more efficient "cooperative"

messaging domain.

d. To simplify many of the complexities common to scripting languages, Master Class developed a *Graphical User Interface (GUI)* with standard objects for CAST to manipulate. These objects include many of the menu-driven utilities common to less powerful authoring tools: standardized data entry fields, on-screen buttons, on-screen menus, etc. Such "objects" offer significant productivity improvements for new users.

e. Another productivity enhancement to CAST, introdûced this year, is its unique *playback function,* which "memorizes" the keystrokes and tasks performed by Subject Matter Experts (SMEs) in order to record functional operations without having to script or program those functions. Using CAST in this manner, *SMEs can produce usable portions of courseware with absolutely no programming expertise.*

f. The next scheduled enhancement to CAST will be *porting to Windows NT* . It will already operate in the NT environment with an appropriate server. WebCast is also under development. It will operate on the World Wide Web and Enterprise Intranets. These improvements will continue to broaden the range of platforms and applications for which

CAST is suited. These also illustrate the vendor's continuing commitment to portability and open systems, and coincides exactly with the ongoing migration strategy for all DOD systems.

7. Is the product suited to the target training environment?

While most authoring products were designed for specific operating systems (like DOS) or popular platforms (like Macintosh or SUN), CAST was clearly compatible with every operating system, platform, and application required. And no storage or presentation media, from touch screen to laser player, seemed beyond its proven capability.

8. Will it work on a network, supporting client-server architectures?

Modern client-server architectures can offer many advantages in training costs and efficiency: central management of a global training network, efficient storage of courseware on a host while presenting training on student terminals, etc.

CAST met this challenge, appropriately supporting stand-alone configurations for small training sites, and the partitioning of various functions on larger networks (e.g., training on some platforms, development and administration on others, etc.).

9. Is it fully featured? If not, is it adaptable? Can users influence the design or functionality of the product? Can they have it tailored to meet their specific needs (or do it themselves), and what costs are involved?

These issues proved significant in my product evaluation. My guidelines mandated that any software purchased must offer all necessary functionality, or offer a means to add features cost-effectively as they are needed. I avoided development products which would require costly customization.

Only nominal functionality was offered by most icon-driven authoring tools. This is in keeping with their one real strength (i.e., their sheer simplicity). Adding needed features would prove to be a matter of changing the baseline product, with concurrence of the vendor, and the associated implementation costs and time lines.

Maximum functionality was available only from true scripting languages. While icon-driven tools offer a limited choice of pre-determined features, a scripting language provides tools for writing actual commands to the system, allowing users to determine exactly what the product does. Such products are initially more complex to master, but the return justifies the beginning user's learning curve.

In my evaluation, CAST quickly surfaced as a fully competent scripting language, infinitely adaptable to suit the developer's and trainee's environment. There are no limits to the templates and shortcuts a user may build and employ. Even the look and feel, interaction and flow, and all the teaching elements of the presentation may be tailored for the maximum training effectiveness.

Note that *no additional licensing or related cost is incurred* in tailoring the CAST product to suit local requirements. The small annual fee for software support provides technical guidance from the vendor, and allows/assists such local adaptations. *Software maintenance support is, in fact, optional,* but I considered it vital.

Features allow CAST to interface with any number of other applications (e.g., one's favorite word processor or graphics tool). If a user desires functionality not currently inherent in CAST, it can be added by accessing other software through the CAST script. As for specific product modifications, discussions with other CAST users confirmed the vendor to be responsive and reliable, accepting product improvement suggestions from users and following through on promises made, often beyond their expectations.

I also verified Master Class routinely accepted trouble calls over the phone and historically provided a satisfactory technical response within 24-hours of notification, and often while customers were still on the phone. On a customer service scale of 1-10, users consistently rated Master Class Corporation a "9" or "10".

10. Will the product satisfy the government's stringent security demands?

Through outstanding security features such as enabling keys, separate authoring and user modules, and built-in management functions, CAST meets every necessary security requirement of the classified military intelligence training environment.

CAST has been in use within the Special-Compartmented-Information Facility (SCIF) at Goodfellow AFB since 1994, and for Senior Year training throughout the AF since '93. CAST has also been licensed for use on Compartmented Mode Workstations, and in other special projects of a classified nature (no attribution).

11. Is the product licensing relatively flexible and cost-effective?

Most product vendors were only willing to sell their product as a full software suite, which is analogous to saying you can only buy Microsoft Word if you also buy

Excel, PowerPoint, and all the rest with it.

Master Class, instead, offers their product in discrete functional modules, enabling users to purchase licenses for only those portions or functions required. Yet, even as a full suite, CAST proved less costly than other fully featured scripting languages (it was about half the price of Gain Momentum, another UNIX authoring system).

CAST is commercially available by terminal count, site, project, or enterprise licensing agreements.

B. PRODUCTIVITY CONSIDERATIONS

The productivity measure of a Computer-Based Training (CBT) development tool is a mathematical ratio of *developer hours* (including overhead items) invested in generating courseware to *presentation hours,* which describes the amount of time required to present an automated lesson to a typical trainee.

In defining an acceptable CBT productivity level, the Air Force determined the established industry standards for production of highly-interactive, graphically-supported courseware with complex learning objectives ranged between 200:1 to over 300:1, depending upon the formula used. The researchers compared this to those productivity figures available from other established users of CAST, and found it

to be within range.

Like any sophisticated software tool, this product does present a learning challenge for new users. While practiced programmers or "scripters" take quite readily to the use of CAST's "English language" based script, those unfamiliar with CBT development or the target training architecture may require one to three months to master this authoring tool. Note that this learning curve compares favorably with the experiences with other authoring tools. "For example, *Accord* required a six-month learning curve. When compared to others, it was determined CAST offered a learning curve comparable to or shorter than *Authorware Pro's* and a relative learning curve (feature for feature) comparable to or moderately longer than *Quest's*, but offered significantly more capability than either." (Carpenter, 1996, p.3) At the same time, it must also be mentioned that certain software utilities offered with CAST allow the initial users in an organization *to tailor and simplify use of the tool* for successive users.

"In any event, CAST was selected for its exceptional strength in all areas except *initial ease of use.* The Air Force accepted the initial learning curve as a necessary trade-off for a quality production, and proceeded with

courseware development in the summer of 1994." (Carpenter, 1996, p.3)

C. SIGNIFICANT ADDITIONAL BENEFITS OF CAST

During product research, the Air Force uncovered significant additional benefits of the CAST authoring tool. It quickly became apparent that CAST was designed to be a potent, highly-efficient tool. In illustration:

1. CAST allows developers to make script changes and view the results "on-the-fly". This offers unequaled opportunity to experiment with and "tweak" courseware, correct errors, and to add to or update a lesson, quickly and easily.

Other products required a major investment in time and trouble just to change a screen display, a test question, or reword text. Modifying scripts required a user to re-compile the program, exit the developer's function, sign back on as a "student", then trudge through the student lesson to see the results (in most cases in lock-step with the normal presentation). If the results were not quite satisfactory, the developer would have to sign back in as a developer and repeat the entire process to "tweak" it again.

2. CAST offered strong efficiencies in "collaborative computing". *Similar to multi-tasking but in the CBT*

training environment, CAST's interaction with other applications on the system is more like a network manager than a baby-sitter. CAST launched queries, issued commands, and boot up/shut down other applications, then went on to the next task without having to wait for a response.

When other authoring tools queried a file or data base as part of the training lesson, the program sat in a "wait state" until the query or task was completed. This is tedious to students and counter-productive in a teaching environment. It also wastes valuable access time on the training station.

3. A most useful element is CAST's Computer-Managed Instruction (CMI) module. That package allows users to effortlessly administer and track the ongoing training. With the push of a button, the CMI module reports on student population, course schedules, etc. *The CMI also allows electronic oversight of remote training sites.*

4. A *major cost benefit offered by the product* is its ability to build a training package around other applications without modifying their source code. Essentially, CAST can be used to build a *non-intrusive training interface* to an application without the involvement of the third-party software vendor. This eliminates the

need for third party consent, associated code re-writes and development costs, additional licenses, etc.

In fact, I learned CAST users do not require a target application's source code or user documents to create a training interface. In one case, CAST was used to train an application for which there was no documentation available.

5. Another strength is its *adaptability to training media and methodologies* anticipated in future training. CAST can handle interactive video disks and touch screens, animation, sound systems and digital video. It has all the requisite features for use in a dynamic multimedia environment.

6. Because the Air Force's target training audience was large (in excess of 1500 initial students worldwide), and the prospective students all had differing backgrounds (branch of military service, experience levels, etc.), another particular feature of CAST appealed to them. To supplement each training module on an as-needed basis, *CAST has the ability to create separate training modules accessible from within an ongoing lesson.*

This offers an effective means to provide additional explanatory material for difficult training objectives, and offers automatic remedial training for the more challenged

students. Entire reference libraries of images and text, sound and graphics can be separately archived and accessed by the student when required.

The Air Force has already realized a significant fallout benefit from this module-building capability, by using it to develop on-line imagery analysis tools. Such innovative uses of CAST allow developers to go well beyond customary CBT design limits.

7. The Air Force's growing understanding of the product design, the future enhancements planned by the vendor, and their recent activity with Defense Intelligence Agency's "System Acquisition & Services Support" contract make it clear that CAST may offer utility far exceeding Computer-Based Training. CAST will eventually lend itself to use in a number of other areas not originally considered, particularly in the areas of:

a. Network Management

When one considers its ability to launch, monitor, control, and halt other applications at the highest levels of an architecture, CAST's potential as a network management tool becomes apparent.

b. Performance Support Systems (PSS)

Offering the means to create a simple user

interface for monitoring and assessing all critical data flow on a system, CAST can be used to assess ongoing student performance, software or system performance, and testing and prototyping.

c. Modeling and Simulation (M&S)

There is a role for CAST in M&S as well. Throughout the DOD, the primary goals of M&S are "planning, cost avoidance, and training". With its ability to interface with and emulate everything from a training system to a war-fighting scenario, CAST users can more effectively plan their undertakings, practice their skills, and dynamically answer numerous "what if" questions on-the-fly, without committing resources or making costly mistakes.

d. Internet & Web Access

With the goal of more effective resource and student management, the DOD is now attempting to electronically interconnect its geographically-separated units, and to integrate its multiple "training data bases". Fortuitously, such a requirement was foreseen during the Air Force's 1992-93 product evaluation; that research highlighted CAST's ability to function across a network, and to tailor, integrate, and remotely access its training data bases. Currently, all of CAST's student management

functions (Computer-Managed Instruction modules) and all its textual functions are fully-supportable across Internet. As mentioned earlier, CAST is developing webCast, which will work with Netscape Navigator and Microsoft Internet Explorer, in order to deliver courseware across the World-Wide-Web. More information on this is provided in Appendix B.

D. SELECTION OF CAST BY THE AIR FORCE

In summer 1993, after a rigorous seven-month investigation into more than forty (40) authoring tools, the CAST product was proposed as the standard development tool for "exportable automated intelligence applications training", by the Integration and National Intelligence Systems Office at Goodfellow AFB, the center of Air Force intelligence training.

During product comparisons, all available commercial and government products were scrutinized for such factors as versatility, "platform portability", ease of use, and cost. Other considerations in their product selection included demands not normally required of authoring tools, including the security needs of their classified courseware, network access to training modules, and global student management from a central facility.

Government budgetary constraints also placed a premium on software versatility; that is, products which could manage imagery, animation, sound, video, dynamic data bases, and a multitude of hardware devices in both the DOS and UNIX environments. Furthermore, as it is a government-related project, the product had to be compliant with all pertinent Federal government and DOD standards for software and its licensing.

CAST was selected by the Air Force after (a) extensive reviews of product documentation, (b) acquisition and licensing guidance from AF Material Command, detailed productivity/performance discussions with established CAST users within and out of the government, (d) three site visits to the vendor facility, (e) building of a prototype training architecture and product, and (f) demonstrations of the prototype throughout the US intelligence community (including training centers and operational units).

Other products simply failed to demonstrate the power and versatility of CAST. Additionally, no other vendor demonstrated, to their satisfaction, such a consistent history of responsiveness to customers in their product development, with such a strong focus on continual product enhancement.

Note also that CAST not only met the government software requirements, but also the more stringent requirements of the DOD Intelligence Information Systems (DODIIS) community. CAST is now being evaluated as a standard for all DOD automated intelligence training applications. Once approved it would become the first commercial product offered as a standard training development tool within the global military intelligence community.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

With the shrinking resources the Navy and DOD are experiencing, it is apparent that CBT and ICW will continue to perform an important function in the Navy's training environment. As the number of personnel and dollars available for travel decrease, the requirement for exportable training drastically increases. Additionally, personnel will not be able to afford to be away from their jobs for extended periods of time due to the reduction of available personnel to cover for them. This demonstrates a greater demand for job cross-training, which requires more personnel to receive the same training.

The evaluation process which resulted in selection of the CAST product was comprehensive. This research included information from existing users, the product vendors, the Air Force, and independent engineering sources, as well as information obtained from the Internet.

Discussions with commercial companies who specialize in CBT development, current users, customers and FLEASWTRACEN personnel helped determine the priorities about needed product functionality, technical concerns, and training

applications.

Phone research and messages and demonstrations by other users helped establish the utility of the products, their level of vendor support, and product strengths/weaknesses. Users were also asked about their ongoing projects, learning curves for the software, applications on their machines, their hardware and media, and other pertinent topics.

> "As CAST gained credibility, an in-depth investigation was conducted by four Air Force representatives (a podium instructor, a systems planner, and both a military and contracted civilian courseware developer). These representatives spent three full days at the vendor facility. The team was given unrestricted access to the CAST software and development labs, and they freely experimented with a development project, Computer-Managed Instruction modules, graphics, and soft-copy imagery.
>
> Following that, a demonstration of the CAST development tool integrated with a target training application was created for evaluation by operational sites and agencies throughout the intelligence community. The same prototype was demonstrated to a global training users' conference hosted by the Air Force in Fall '93. Feedback from the research efforts and the many evaluators was consolidated into several short papers and disseminated for discussion and to invite more feedback.
>
> Finally, after continuing political challenges by other military services and vendors, and after extensive debate by financial, programming, training, and advisory personnel, actual training development with CAST began in earnest in summer of 1994. Currently, the product is under evaluation as a candidate for satisfying the second and broader requirement, that of converting other existing intelligence skills

training courseware to an automated and exportable format." (Carpenter, 1996, p.5)

CAST successfully met all of the selection criteria, along with the training needs of FLEASWTRACEN and the purchasing requirements of DOD.

B. RECOMMENDATIONS

I have developed the following recommendations based on the information contained in this paper:

1. Purchase CAST Software

The purchase of the CAST software requires a minimal initial investment. Master Class Corporation supports their product 100% and virtually guarantees customer satisfaction. If the product does not perform up to a customer's expectations, no further investment is required. Additionally, cost of the product and licensing is a one-time fee and is dependent on the scope of the project and the economic benefit to the customer.

Although the initial learning curve is high, as compared to icon-driven tools, my research concluded that non-programmers could master the language in one to three months. Once the language is mastered, the product will fulfill all of the items on the "wish list" and more. The capabilities of CAST are virtually limitless and will be

able to expand and grow with the training requirements.

2. Training via a Network

I also recommend that the training provided by FLEASWTRACEN be transmitted via a network. The Air Force conducts their training on-line. This enables them to make changes to their lessons and send the them to all of their customers at once. This is very cost-effective reducing not only the author's time required to maintain the lessons, but also the cost of producing CDs. Lesson presentation on-line is simplified due to the ease of portability CAST offers.

The lessons already produced by Mandarin can be transferred to CAST with relative ease since CAST is able to control the graphics, audio and visual applications. Since the lessons are already created, the process would just require the material to be input using the scripting language.

3. Purchase a UNIX Server

In addition to the initial investment of the cost of the software, I recommend that FLEASWTRACEN also purchase a UNIX server for the authors' network. The UNIX environment provides a greater degree of flexibility to the authors as compared to an icon-driven environment. A UNIX server for the students' network would also be beneficial, providing

more options to the student, but that is not imperative at this time.

4. Consideration of LearnStar Corporation

Due to the simplicity and routine classes, I recommend that LearnStar Corporation be considered to produce software for classes given in the boot-camp or "A" school environments. These lessons are repetitive and require very few updates. (i.e. A lesson on the rank structure.) By producing these lessons in the LearnStar environment, they will be more exciting and require interaction from the students. Additionally, the need for an instructor is eliminated because this role could be filled by a "company commander," who would supervise the activity, and is normally present anyway.

C. TOPICS FOR FURTHER RESEARCH

There are many avenues for additional study in the Computer-Based Training arena. An analysis similar to this thesis should be completed at least every two years to ensure that the current software is effectively meeting the training requirements, and is not "out-of-date."

Additionally, further research should be conducted to investigate the alternatives to training via interactive courseware. Some of the alternatives would be "distance

learning" or Interactive Technical Manuals (IATMS).

LESSON STATUS

PART:	
LESSON:	
NAME:	
AUTHOR:	
START:	
COMPLETE:	

DEVELOPMENT STAGE	*Days*	*Turn-In*	*Returned*	*Due Date*	*Actual Date*
CONTENT OUTLINE					
Develop Exercise Scenario	2d				
T/L Review	.5d				
QA Review	.5d				
STORYBOARD					
Develop Storyboard	6d				
T/L Review	1d				
SME Review	1d				
QA Review	1d				
Incorporate Comments	1d				
T/L Final Review	.5d				
Turn-In Graphics					
Export Storyboard	.5d				
Storyboard Complete					
IMPLEMENTATION					
Author First three Keypoints	5d				
Inhouse Audio Complete					
First three Keypoint Graphics Complete					
T/L Review	1d				
Incorporate Comments	1d				
Author Remaining Keypoints	9d				
Remaining Graphics Complete					
T/L Review	.5d				
SME Review	.5d				
QA Initial Review	1d				
Incorporate Comments/Revise narrative	2d				
T/L Review	.5d				
QA Final Review	.5d				
Pro-Audio	1d				
Implementation Complete					
POST PRODUCTION					
Pro-Audio complete					
Down Load to DiskPack	.5d				
Author Lesson Review	.5d				
Q.A. Lesson Review	.5d				
Lesson Complete					

LESSON STATUS

PART:	
LESSON:	
NAME:	
AUTHOR:	
START:	
COMPLETE:	

DEVELOPMENT STAGE	*Days*	*Turn-In*	*Returned*	*Due Date*	*Actual Date*
CONTENT OUTLINE					
Develop Exercise Scenario	3d				
T/L Review	.5d				
QA Review	.5d				
STORYBOARD					
Develop Storyboard	7d				
T/L Review	1d				
SME Review	1d				
QA Review	1d				
Incorporate Comments	1d				
T/L Final Review	.5d				
Turn-In Graphics					
Export Storyboard	.5d				
Storyboard Complete					
IMPLEMENTATION					
Author First three Keypoints	7d				
Inhouse Audio Complete					
First three Keypoint Graphics Complete					
T/L Review	1d				
Incorporate Comments	1d				
Author Remaining Keypoints	11d				
Remaining Graphics Complete					
T/L Review	.5d				
SME Review	.5d				
QA Initial Review	1d				
Incorporate Comments/Revise narrative	2d				
T/L Review	.5d				
QA Final Review	.5d				
Pro-Audio	1d				
Implementation Complete					
POST PRODUCTION					
Pro-Audio complete					
Down Load to DiskPack	.5d				
Author Lesson Review	.5d				
Q.A. Lesson Review	.5d				
Lesson Complete					

LESSON STATUS

PART/LESSON	
NAME:	
AUTHOR:	
START/COMP	

DEVELOPMENT STAGE	Days	Turn-In	Returned	Due Date	Actual Date
CONTENT OUTLINE					
SME/Q.A. Service	.5d				
Develop Content Outline	5d				
SME/Q.A. Service	.5d				
T/L Review	.5d				
SME Review	.5d				
QA Review	.5d				
Incorporate Comments	.5d				
T/L (Check incorporation of comments)					
Complete Content Outline					
STORYBOARD					
Develop First three Keypoints	5d				
SME Service	.5d				
T/L Review	.5d				
Incorporate Comments	.5d				
Develop Remaining Keypoints	8d				
T/L Review	3d				
SME Review	2d				
Q.A. Review	3d				
Incorporate Comments	2d				
T/L Final Review	1d				
Turn-In Graphics					
Export Storyboard	.5d				
Storyboard Complete					
IMPLEMENTATION					
Author First three Keypoints	7d				
Inhouse Audio Complete					
First three Keypoint Graphics Complete					
T/L Review	1d				
Incorporate Comments	2d				
Author Remaining Keypoints	15d				
Remaining Graphics Complete					
T/L Review	1.5d				
SME Review	1d				
QA Initial Review	1.5d				
Incorporate Comments	4d				
T/L Review	.5d				
QA Final Review	.5d				
Professional Audio	1.5d				
Implementation Complete					
POST PRODUCTION					
Down Load to DiskPack	.5d				
Author Verify Lesson	.5d				
QA Lesson Acceptance	.5d				
Post Production Complete					

LESSON STATUS

PART/LESSON	
NAME:	
AUTHOR:	
START/COMP	

DEVELOPMENT STAGE	Days	Turn-In	Returned	Due Date	Actual Date
CONTENT OUTLINE					
SME/Q.A. Service	.5d				
Develop Content Outline	6d				
SME/Q.A. Service	.5d				
T/L Review	.5d				
SME Review	.5d				
QA Review	.5d				
Incorporate Comments	.5d				
T/L (Check incorporation of comments)					
Complete Content Outline					
STORYBOARD					
Develop First three Keypoints	6d				
SME Service	.5d				
T/L Review	.5d				
Incorporate Comments	.5d				
Develop Remaining Keypoints	10d				
T/L Review	3d				
SME Review	2d				
Q.A. Review	3d				
Incorporate Comments	2d				
T/L Final Review	1d				
Turn-In Graphics					
Export Storyboard	.5d				
Storyboard Complete					
IMPLEMENTATION					
Author First three Keypoints	9d				
Inhouse Audio Complete					
First three Keypoint Graphics Complete					
T/L Review	1d				
Incorporate Comments	2d				
Author Remaining Keypoints	19d				
Remaining Graphics Complete					
T/L Review	1.5d				
SME Review	1d				
QA Initial Review	1.5d				
Incorporate Comments	4d				
T/L Review	.5d				
QA Final Review	.5d				
Professional Audio	1.5d				
Implementation Complete					
POST PRODUCTION					
Down Load to DiskPack	.5d				
Author Verify Lesson	.5d				
QA Lesson Acceptance	.5d				
Post Production Complete					

WEBCAST

An Unusual CGI Scripting Language

Barry K. Moritz, Ph.D.
Master Class Corporation
Parkway 2 Suite 100
2697 International Parkway
Virginia Beach, Virginia 23452
(757)427-5050 email:mc2@exis.net

ABSTRACT

What the world needs is..... "a powerful but extraordinarily easy to use and productive open system architecture package that allows inexpensive development and delivery of highly responsive, multimedia based, dynamic, and intelligently interactive multi-page applications (including training) for the world wide web through a secure medium bandwidth linkage between browser and server" (whew!). This is a dream that time will likely make a reality (or cause to become moot due to availability of high speed links). But, for the moment, there is no such thing. Nevertheless, there is ample evidence for similar systems in the X Window environment. One such product is Master Class Corporation's CAST® system. The realities of paradigm differences between LAN/X architecture and web architecture that are caused by differences in bandwidth, demand levels, and original intent make the transition of an X Window - based application to the web an interesting exercise.

KEYWORDS

Authoring System, Computer Assisted Learning (CAL), Computer Based Training (CBT), Common Gateway Interface (CGI), Interactive Multimedia, Internet, Intranet, Scripting Languages, world-wide-web

INTRODUCTION

There is a pent up demand for the delivery of intelligent interactive applications and effective multimedia based training on both the world-wide-web and private intranets. Unfortunately, the tools for developing such applications demand the heavy and expensive involvement of computer programming experts who have the training and/or experience necessary to deal with low level shell scripting languages, "system level" languages such as C and Java. and other somewhat higher level languages such as TCL/TK and Perl. Higher level languages such as Microsoft's BASIC are, by design, being kept for use on proprietary operating systems (Microsoft's NT) and are therefore not available for the majority of web servers (more than 80% at this time are on UNIX-like systems).

CAST® is a system designed for the development and delivery of highly interactive programs that involve intelligent bi-directional communication of graphic, text. and control information (discourse) between a person and a machine. It has found a natural niche in the world of Computer Aided Learning (CAL) and Computer Based Training (CBT). The CAST system was the first commercially available CAL/CBT Authoring System for UNIX-based systems and enterprise networks (1984), and is still most often used in that capacity. The system is built around an interpreted Very High Level Language (VHLL) also called CAST.

The CAST system has already been made operational on standalone systems and heterogeneous networks involving more than 50 different operating systems (including most flavors of UNIX, VAX/VMS, and MS-DOS). It is available for use with character-based terminals as well as graphics-equipped units such as X Terminals, workstations, and X servers (under Motif and OPEN LOOK), and VGA or TIGA equipped MS-DOS PC machines.

CAST is a UNIX-based open systems architecture product from stem to stern. The port was **from** UNIX **to** DOS and VAX/VMS, not any other way. As a paid-up member of the open systems community, CAST is designed to incorporate other applications and processes accessible on the distributed open systems network. The term "incorporate" doesn't simply mean that CAST just launches another application. Rather it also places itself between the application's user I/O and the user. It can

©1996 Master Class Corporation
Cast and MC² are registered trademarks of Master Class Corporation. All other company and product references may be trademarks of or otherwise proprietary to the respective companies with which they are associated.

monitor the traffic, intercept the traffic, copy and reformat or store traffic for later use, and otherwise completely control an application as if CAST itself were the terminal. This works for both "stdio" applications and X Window clients.

The emergence of the world-wide-web as a global standard technology offers yet another environment for CAST. As a result, Master Class is again enhancing the CAST product to be operational as a high productivity CGI server scripting language for the internet and intranets. The name "WebCAST" has been used to identify this version of CAST. A prototype of WebCAST is now being demonstrated.

As with other CGI scripting language products, WebCAST will be able to create html pages "on the fly" based on information obtained from anywhere within the networked system.

The baseline of the html technology's static presentations and the architecture of web servers that is necessary to ensure adequate server throughput are significant challenges to providing CAST's historical high levels of author productivity in an environment of intelligent multimedia discourse.

The balance of this paper discusses some of these challenges and the approach taken by Master Class to meet them. The current capabilities of CAST are used to establish the baseline of the product's characteristics.

EASE OF USE

The question of whether it is easier to use a language or a point-and-click or menu-based development tool is probably nowhere near as hotly debated as in the CAL/CBT community. Historically, CAL/CBT courseware developers (authors) have resisted the use of languages because they were difficult for non-programmers to use and/or because productivity was inadequate.

This is just the problem that CAST was designed to solve. The CAST language is not a died-in-the-wool programmer's dream - rather it is a more plebeian high level workhorse based on English. It is designed to be used by pseudo programmers with, perhaps, only an occasional bit of help from more sophisticated programmers.

CAST has its roots in the venerable PILOT CBT authoring language - a language which, despite its previous tendency to produce boring CBT, was judged easier to use than any other language. But CAST's trunk is more modern - with the inclusion of some simple Object Oriented Programming (OOPS), a nice range of image-oriented graphical capabilities (including some animation), more sophisticated facilities to evaluate student responses other than 1-3 or a-c, a capability for incorporating and directly communicating external programs as extensions of a script, and an ability, like LISP, to dynamically create its own instructions as appropriate to the situation (this is where, however, a more experience programmer would come in to the picture).

Prior to the release of the first version of CAST in 1984, Master Class undertook a study to determine how quickly non-programmers could become productive with CAST. Using K-12 teachers with no programming (and little computer) experience as subjects, it was determined that a first simple but pedagogically correct five minute CBT course could be developed within two weeks. Just last year the United States Air Force Air Education and Training Command (AETC) had a similar experience on a major CBT development effort. In their case, a courseware developer with no prior CAST experience was efficiently generating production-quality courseware within a month. That organization also recently received a prestigious award for the application of technology to training; the technology was CAST.

But perhaps the most striking example of a non-programmer's use of CAST is the simulation of the Motif window manager interactive graphic operations of drag and drop and window resizing. Undertaken as part of a larger CBT course, the simulation performance was about the same as the actual window manager software, so that you could not tell the difference between them. The simulation worked on X Terminals as well as the networked workstation consoles.

So it would seem clear that the CAST language's design objective of "ease of use" by non-programmers has been met until now. Yet the world wide web environment brings a new set of complexities that must be addressed. The two primary issues are state preservation and access control.

State Preservation

The world wide web is a stateless environment oriented to a single page of html information. Information generated from one page is not easily or automatically preserved for use with later pages visited within a session. To accomplish the transfer of state information from earlier session pages to later pages one must now do some programming in one of the available CGI scripting languages (e.g., Perl, TCL, C, shell scripting languages). This is work for experienced programmers and is usually outside the capabilities of most publishers of web pages.

However CAST scripts, as currently written, provide variables and system information which ensure the simple and immediate availability of such information across all the "pages" in a session. WebCAST will offer this same capability - a seamless, automatic preservation of state information across multiple pages in a session. CAST authors will be able to use simple variables to store and retrieve such cross-page and historical information. This session information will be automatically saved when a new page is transmitted to the web browser and restored upon receipt of information from that page. In essence, the state preservation issue would be a non-issue for CAST users.

Access Control

The ability of a web browser to cache pages for later revisiting offers a unique challenge to ensuring the control of page sequencing and access - both within a session and over longer periods of time. Again, as with the issue of state preservation, a user of currently popular CGI scripting languages would encounter some serious programming.

WebCAST will handle this problem for the author by requesting the author only to indicate, at any time, which "pages" (actually labeled sections of scripts) are available for use within the current session. Requests and data from out-of-session pages or out of sequence pages for the current session will not be honored. The author may, however, cause such requests to be directed to a standard "error" response. In other words, CAST will provide built-in support to controlled navigation.

Without such a capability to control access to pages, it would be impossible to produce training materials that guarantee the student's exposure to all information in the desired order or that take advantage of alternate learning strategies and materials based upon actual student performance.

Ease of Use Versus Ease of Learning

There is little doubt that virtually any language has a steeper learning curve than an "intuitive" point-and-click user interface. Yet the above presented information would lead one to believe that CAST is extraordinarily easy to learn and that it can be constructively applied even by non-programmers. Given the achievement of a simple method for state preservation and sequencing control, the same might be said about WebCAST.

However this leaves open the question of whether an easy-to-learn system is also easy to use (or vice versa, whether a system more difficult to learn is also more difficult to use). This is discussed in the following section.

PRODUCTIVITY

There is ample evidence throughout the history of toolmaking that professionals choose tools for which skills may be harder to acquire, but which produce better and quicker results. In other words, a professional tool is claimed to be easy to use if its application results in higher levels of productivity and product quality.

Productivity is the name of the game in software and courseware development. As should be well known to all, the cost of software development far exceeds the cost of the software system upon which the development is based. Compilers and/or interpreters for C, Java, and other languages are either free or cost at most a few thousand dollars. That cost is typically less than the weekly cost of an experienced programmer - and far less than the cost of training a less experienced programmer. So, if one can increase development productivity by 20%, the developers experience a commensurate reduction in the cost of the application.

Such thinking is used to select a development product most applicable to and productive for any specific job, subject usually to rightful concern about the experience of the development team with the product. The same thinking is often behind the selection of a point-and-click or menu-driven authoring system over a language based system. It is assumed that a language based system will, by its developer interface alone, be less productive than a point-and click system. As Master Class has proven in documented experiences of its customers, such assumptions are often fallacious - particularly for larger projects or multiple smaller projects sharing common elements.

Productivity has been a hallmark of the CAST product throughout its history. If we measure productivity as the number of developer hours necessary to produce an hour of courseware, CAST's productivity has been demonstrated (when used on projects and not as an occasional dabble) as anywhere from at least 20% better to more than twice that you can expect from other leading point-and-click CBT authoring products. And this productivity applies to whatever delivery and presentation method or technique is chosen - not to only a subset of possibilities made available by a point-and-click interface.

The productivity gains are not limited to inexperienced programmers. As an interpreted VHLL, CAST offers efficiency and productivity unavailable in compiled languages and lower level languages like Java. In the hands of an experience programmer, CAST can offer

productivity gains over C by up to a factor of 40 - particularly when it is used for interaction with the user and not for things better reserved for C.

Master Class believes that the CAST VHLL, although extraordinarily well suited for developing the human/machine interface elements of an application, is not a universal development solution. Other elements such as data base subsystems, expert system components, calculation intensive routines, or interrupt driven peripheral control code are also part of a complete application, and for those CAST is not well suited.

Therefore CAST is designed to incorporate and use such elements, whether they are Commercial Off The Shelf (COTS) or custom routines, and no matter in what language they are written. This allows an application developer to achieve the highest productivity levels in each element of a total application and to bring the elements together under a CAST VHLL script.

In the web environment, CAST can therefore act as a translating interface (a "filter") between legacy software and the web user. It can equally act as a common interface gateway to a collection of such legacy software, whether that software is X based or terminal based.

The problem of throughput capacity will have to be considered, however. Most web servers are small machines with precious few resources to commit to multiple resident processes. It is for that reason that the CGI specification was written as it is and why the web publishing consists predominantly of creating static pages. The restrictions will begin to lift as new server products such as FastCGI[1] and higher performance machines and internal networks become more prevalent. For now, the prototype version of WebCAST is small, loads rapidly, can be made to leave quickly, and therefore requires few system resources. CAST cannot remain resident under the rules of the CGI interface, and that precludes the exercise of some of CAST's facilities for using external applications. However, nothing precludes leaving CAST resident, so the use of technology such as FastCGI can allow CAST to exercise its full potential. Master Class is now working on mechanisms to allow continued residency without violating CGI rules.

PORTABILITY

As previously stated, the CAST system has run on more than 50 different operating systems and is equally at home on character-oriented terminals, MS-DOS VGA or TIGA displays, and any machine on which an X server is available (X terminals, UNIX and UNIX-like workstations (e.g., HPUX, IRIX, BSD, Linux, SCO, SOLARIS, SUNOS, OSF1,....), MS-Windows and NT boxes, etc., etc.). It is therefore one of the most portable sophisticated pieces of software now in existence.

CAST scripts can run in any computer to which the CAST system has been ported. Like Java, CAST scripts can be made completely independent of the system on which they run. Alternatively, CAST allows an author to sense the system on which a script is being run and, like the html-oriented web browser, to allow alternate presentations based upon the system's capabilities. Thus there are some similarities between Java and CAST, but the CAST language is at a **much** higher level.

In this day of distributed systems it is important to be portable not only in the sense of where the program executes, but with what terminal device the user accesses the program. The original version of CAST made it possible for a single script to be equally usable on nearly every character-based terminal available - even when a system had mixed terminal models. Through this same capability, even terminals with some graphics capabilities were handled (although we never did get to implement NAPLPS before that became a moot point).

A prototype version of WebCAST (the CGI scripting language version of CAST) now only requires a CAST user to be exercising one of the available web browsers such as Netscape Navigator, Microsoft Internet Explorer, or Mosaic.

The use of html and the limited bandwidth available to the typical web user are constraining factors not experienced with other display mechanisms now used by CAST. Web users and publishers are well aware of the static nature of html documents and the excessive time often taken to load a page.

The throughput problem is often ameliorated or removed in private intranets when high speed local area loops are used. In these cases, the net owner is able to choose between the current version of CAST and WebCAST - both of which share the same language base.

Java-capable browsers are also now available. Therefore Java scripts can be embedded within html documents to enhance the dynamicism of the displayed page and, subject to security considerations, to perform other local computations and data access. A similar functionality is available for some browser plugs-ins such as TCL and XFast.

WebCAST is able to create such pages with embedded Java code on the fly, by transfer of existing html files, or by a

combination of these. Future versions of WebCAST are expected to enhance the coupling between Java and CAST so that CAST's full suite of interactive graphics and object placement on the display will become available.

CAST (and WebCAST) scripts are best developed for now on UNIX hosts using the X Window system, although they can be developed on character-based terminals or on MS-DOS systems. The CAST development system for X Window uses four windows - two for controlling the development process, one for script editing, and one to show what the user will see (the student display window). The student display window for WebCAST is generated by the developer's browser. For CAST it is generated by an element of the CAST development code.

POWER AND FLEXIBILITY

In short, the CAST language is powerful enough to be used as a general purpose programming language. In addition to virtually unlimited testing and branching, it supports a complete suite of file and device access including facilities for searching files for desired text fragments. Its text string processing facilities are quite powerful, which enables it to fully parse strings entered by the user or made available through other processes or files. It offers the basic mathematical functions (addition, multiplication, division, exponentiation, and term grouping) as well as the Boolean operations of NOT, AND, OR, and XOR.

Each CAST script (a file) can be referenced by any other CAST script as a subroutine. A single copy of such a "subscript" is all that is needed, thereby minimizing storage requirements and maintenance costs. Scripts can also be "included" in other scripts. Such included scripts usually contain elements which establish an application's look and feel. To modify the look and feel for an entire application, it is necessary only to modify the single "included" script.

Also present are commands for the location, sizing, and display of text in selectable fonts, imagery, and icons. Display objects can be made to replace existing areas of the display or to modify them with OR, AND, and XOR processing. Areas of the display can be protected from change, and text or other glyphs can be placed on top of graphics.

Through these display related commands, it is not necessary to use a completely separate image for each variant of a layout. This approach, although highly desirable to reduce network bandwidth requirements, is diametrically opposed to the static display approach used by "vanilla" html. Master Class expects that judicious use of Java scripts and/or client-resident Java, TCL, or other uploadable programs will span the differences so that minimum bandwidth will still be required even on the internet.

Areas of the CAST user display window can be made active to mouse clicks, double clicks, and drags - each area announcing such mouse actions with a unique message. The areas can be associated with visual controls or can be transparent. If desirable, the mouse motion itself can be dynamically followed.

Given that all these above capabilities are available in an English-like VHLL, an author is capable of cost effectively achieving virtually any look and feel. To quote a phrase expressed by one Master Class customer, "you are only limited by your imagination."

Whereas the computational, text, and file processing capabilities remain unchanged from CAST to WebCAST, the nature of html makes it impossible to translate CAST's multimedia capabilities directly to WebCAST without extensions such as Java. Several mechanisms for making this translation as seamless as possible are now under consideration.

APPLICABILITY

Master Class has focused on the use of CAST in the training and computer assisted learning markets as it believes training involves intelligent bi-directional discourse rather than only tutorial, page-turning monologue. To support this application, Master Class has introduced what it believes to be the only commercially available built-in Computer Managed Instruction (CMI) capability. CAST's CMI helps build complex courses based upon lower levels (e.g., units, lessons, and sessions) with prerequisite, performance gateways, and other evaluation logic. It allows a "registrar" to create classes, assign courses to classes (or vice versa, if you'd rather), and assign students to classes or directly to courses. It captures desired student performance information in a data base which can be exported into a full-blown DBMS. It also allows an assigned instructor to modify a student's schedule.

When it comes to Intelligent CAL, CAST's ability to incorporate, communicate with, and fully control external applications allows any expert systems or DBMS packages to become CAST helper functions. CAST furthermore helps an author more easily deal with external "helper" programs through its "agent" interface. Several CAST extensions, including Concurrent X Control (CXC) and the GUI agent already use this mechanism.

VAPORWARE AND THE FUTURE

It must be made perfectly clear that, despite the successful demonstrations of a WebCAST prototype, WebCAST is not yet an announced product. The demo does show a WebCAST generated game with changing graphics (tic tac toe), a short tutorial (cultural diffusion, or why English isn't spoken in more places), and a developer's tool (for viewing image files from a directory of files using an html form, the "ls" command, a dynamically generated html menu and an IMG tag). Whereas WebCAST already integrates with the CGI information pipeline protocols and conventions (including browser and server generated information), much of this CGI-related functionality of the WebCAST prototype is still written in the CAST language. That code will be rewritten in C with a simpler author interface before product release.

Other elements to enhance author productivity in a web environment will be added before product release. For example, we will be duly considering the wealth of html page creation and editing software already available on the market, and introduce capabilities to take advantage of the results of these other systems.

We'll also be working to ensure maximal language compatibility between scripts developed for X and those developed for html. Both X technology and web technology will co-exist on intranets (and, given Broadway or something like it, probably merge). We intend to be ready.

For now, Master Class intends to focus on the server side, where we can expect to make the most impact. We therefore plan to use Java and other already-accepted and highly available technology as the browser helpers. The long range direction in which we move, however, is based upon the portability of the C code from which the CAST system is built and the portability of CAST scripts themselves. A future stop on the road therefore offers the possibility of "CASTScript" (i.e., web client resident CAST).

REFERENCES

1. FastCGI: A High-Performance Web Server Interface, Open Market Inc. April, 1996: http://www.fastcgi.com/kit/doc/fastcgi-whitepaper/fastcgi.htm

LIST OF REFERENCES

ASK International Home Page. (Netscape Web Page).

Bandrowski, Paul, "Throw Out the Textbooks: Multimedia Training Takes Off." *Corporate Computing,* Vol. 1, No. 5, November 1992, p 206.

Baxter, Barbara, "State Tech at Memphis Addresses Community Literacy Needs; The Workforce Training Center, a Three-Year Pilot Program at State Technical Institute." *THE (Technological Horizons in Education) Journal,* Vol. 19, No. 8, March 1992, p 54.

Bjorner, Susanne, "The Virtual College Classroom; Online Instruction Programs." *Link-Up,* Vol. 10, No. 4, July 1993, p 21.

"Building a Corporate Intranet." On-line Seminar. Wordmark Associates. (Netscape Web Page).

Carpenter, Ron, MSgt, U.S. Air Force, "Product Comparisons." *CAST Authoring Software; Product Selection & Productivity Levels,* April 1996.

"Conversations with the Top Five.", *CBT Solutions,* Jan/Feb 1996.

"Computer Based Training - New Multimedia Product." *Training Media Review,* Nov/Dec 1995.

Computerprep Home Page. (Netscape Web Page).

Coye, Ray W., and Stonebraker, Peter W., "The Effectiveness of Personal Computers in Operations Management Education." *International Journal of Operations and Production Management,* Vol. 14, No. 12, 1994, pp 35-46.

d'Vinci Interactive Computer Based Training Home Page. (Netscape Web Page).

Electric Book, Inc. Home Page. (Netscape Web Page).

Gery, Gloria J. "Twenty Years of Interactive Learning." *CBT*

Solutions, June-July 1995.

Group Performance Systems Home Page. (Netscape Web Page).

Hall, Brandon, "Cost Effective Computer Based Training." *Multimedia Training,* Vol. 2, No. 4, July-August 1995.

Hall, Brandon, "ROI and Multimedia Training." *Multimedia Training Newsletter,* 1996. (Netscape Web Page).

Interactive Development Services Home Page. (Netscape Web Page).

Kalman, Richard, "Future Classrooms: A Personal Vision." *The Mathematics Teacher,* Vol. 87, No. 7, October 1994, pp 486-487.

Kaplan, Randy and Rock, Denny, "New Directions for Intelligent Tutoring; Intelligent Tutoring systems." *AI Expert,* Vol. 10, No. 2, February 1995, p 30.

Lager, Darrel and Koopman, Sue, "The World-Wide-Web as a Medium for Presenting LLNL Training Courses." (Netscape Web Page).

Leidner, Dorothy E., and Jarvenpaa, Sirkka L., "The Use of Information Technology to Enhance

Leonard, Bill, "Technology Has Changed the Way We Train." *HR Magazine,* April 1996.

Lipp, Kim Acker, "Web Based Training Matures." (Netscape Web Page).

Magel, Mark, "Comparative Review of Authoring Tools By Mark Magel." (Netscape Web Page).

"Management School Education: A Theoretical View." *MS Quarterly,* Vol. 19, No. 3, September 1995, pp 265-291.

McManus, Thomas Fox, "Delivering Instruction on the World Wide Web." (Netscape Web Page).

Moranville, April, ICW Division Officer, FLEASWTRACEN, Memorandum dated 9 February 1996.

Moritz, Barry K., Ph.D., Master Class Corporation Founder, Chairman and Chief Executive, Interview on 2 October 1996.

Nichols, Greg, "Formative Evaluation of Web Based Training." (Netscape Web Page).

Nichols, Greg, "Greg's Web Based Training Place." (Netscape Web Page).

Perelman, Lewis P., "Can Technology Effectively Replace Human Teachers?" *Computer World,* October 8, 1990, p 25.

Reinhardt, Andy, "New Ways to Learn." *Byte,* Vol. 20, No. 3, March, 1995, p 50.

Sheldon, Paul, "Using Stories in Multimedia Training." *Trainer's GOLD,* 1996.

Shinkins, Sadie, "Using Computers to Teach Project Management." *Journal of Management Development,* Vol. 14, No. 7, 1995, pp 4-14.

Strategic Management Group, Inc. Home Page. (Netscape Web Page).

Summary of Interactive Courseware Interim Status Report brief, FLEASWTRACEN, Undated.

Sybase Gain Momentum, "Overview", September, 1994.

Tarragon Training International Home Page. (Netscape Web Page).

"The Human Resource Software & Computer Based Training Library." HR Press. (Netscape Web Page).

University Online Home Page. (Netscape Web Page).

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, CA 93943-5101	2
3.	Bruce R. Linder, CAPT, USN FLEASWTRACEN 32444 Echo Lane, Suite 100 San Diego, CA 92147-5199	4
4.	Barry K. Moritz, Ph.D. Master Class Corporation Parkway 2, Suite 100 2697 International Parkway Virginia Beach, VA 23452	1
5.	Laura J. Himmelberg, LT, USN 809 Beaver Creek Court Chesapeake, VA 23322	2
6.	Ron Carpenter, MSgt, USAF Headquarters Air Education & Training Command Technical Training Directorate 244F St. East Suite 2 Randolph AFB, TX 78150-4321	1
7.	Professor Gordon Louvau Naval Postgraduate School Monterey, CA 93943-5101	1
8.	David Matthews, COL, USA, Retired Naval Postgraduate School Monterey, CA 93943-5101	1
9.	Dr. Doug Moses Naval Postgraduate School Monterey, CA 93943-5101	1